Vision-Based 2D and 3D Human Activity Recognition

Holte, Michael Boelstoft

Publication date:
2012

Document Version
Early version, also known as pre-print

Vision-Based 2D and 3D Human Activity Recognition

A Ph.D. dissertation
by
Michael B. Holte

Department of Architecture, Design and Media Technology
Faculty of Engineering and Science
Aalborg University, Denmark
E-mail: boelstoft@gmail.com

November 2011

This dissertation was submitted in November 2011 to the Faculty of Engineering and Science, Aalborg University, Denmark, in partial fulfilment of the requirements for the Doctor of Philosophy degree.

The following adjudication committee was appointed to evaluate the thesis. Note that the supervisor was a non-voting member of the committee.

Reader Ian Reid, Ph.D.
Department of Engineering Science
University of Oxford
Oxford, United Kingdom

Professor Adrian Hilton, Ph.D.
Centre for Vision, Speech and Signal Processing
University of Surrey
Guildford, United Kingdom

Associate professor Claus B. Madsen, Ph.D. (committee chairman)
Department of Architecture, Design, and Media Technology
Aalborg University
Aalborg, Denmark

Associate professor Thomas B. Moeslund, Ph.D. (supervisor)
Department of Arcihtecture, Design, and Media Technology
Aalborg University
Aalborg, Denmark

Abstract

Automatic analysis and recognition of human activities facilitates many applications, and thus has received great interest from both industry and research communities. The focus of this thesis is on video-based human activity recognition, *i.e.* automatic analysis and understanding of activities performed by people and recorded by video. Different kind of activities are considered, from one and two arms gestures (*e.g.* point, wave, clap etc.) to full-body actions (*e.g.* walk, run, jump etc.).

The activities are recorded by standard color cameras, multi-view camera setups and time-of-flight (ToF) range cameras, enabling analysis of both 2D and 3D video data. The 2D image data recorded by standard color cameras captures both activities performed in simple scenes with controlled settings (*i.e.* one actor, steady camera, simple and clean background, and low variation in scale, rotation, viewpoint and illumination) and complex scenes with unconstrained settings (*i.e.* multiple actors, moving camera, background clutter, and high variation in scale, rotation, viewpoint and illumination). For acquisition of 3D data both direct 3D imaging devices (ToF range cameras) and 3D reconstruction from multiple camera views are applied, to explore challenges of different quality of 3D data and the advantages of each technology.

The use of both global and local image features are investigated for activity recognition. A global feature and key-frame based approach is presented to recognize arm gestures in simple scenes. The method extracts a set of characteristic poses and describes them by their motion resulting in motion primitives. A probabilistic edit distance is used to classify a sequence of motion primitives as a gesture. This 2D recognition process is extended into a view-invariant recognition of arm gestures by use of a ToF range camera that generates 3D data and allows for a 3D equivalent of motion primitives.

For recognition of full-body human actions in complex scenes an approach based on detection of spatio-temporal interest points (STIPs) and local description of image features is presented. Robust and selective STIPs are detected by applying surround suppression combined with local and temporal constraints. The approach is especially robust to camera motion and background clutter, where other detectors fails, and improves the performance by detecting more repeatable, stable and distinctive STIPs for human actors, while suppressing unwanted background STIPs.

To give the reader an overview of recent developments in human activity recognition a survey is presented, which reviews and compares recent approaches for multi-view human 3D body modeling, pose estimation and human activity recognition, and discusses the application domain and the associated requirements. To compare the proposed methods, a qualitative assessment of methods which cannot be compared quantitatively is given. Additionally, some prominent 3D pose estimation techniques are analyzed for application, where not only the performed action needs to be identified but also a more detailed description of body parts and joint configuration.

Based on the findings of the prior work on 2D and 3D human activity recognition, the idea of STIP detection and local description of image features is expanded to the 3D domain to generate a more robust and descriptive 3D representation of human actions, which more efficiently deals with the problems of viewpoint changes and occlusion. 4D

(3D space + time) STIPs are detected in multi-view images and novel local 3D motion descriptors, Histogram of Optical 3D Flow (HOF3D), are extracted from estimated 3D optical flow in the neighborhood of each 4D STIP and made view-invariant.

Finally, an approach for automatic foreground segmentation and shadow detection is designed and implemented. Foreground segmentation is one of the most used preprocessing steps for many computer vision algorithms to extract regions of interest, *e.g.* for activity recognition, while the impact of shadows is a notorious problem in computer vision. The work explores a multi-stage approach for foreground segmentation and shadow detection. Firstly, a bottom-up architecture using a novel technique based on gradient and color models is presented for separating chromatic moving cast shadows from detected moving objects. Secondly, a top-down architecture based on a tracking system using mutual object-shadow information is developed, in order to enhance the chromatic shadow detection.

Resume

Automatisk analyse og genkendelse af menneskers bevægelser muliggør et væld af applikationer og har derfor opnået stor interesse fra både industrien og forskningsmiljøer. Fokus for denne afhandling er analyse og genkendelse menneskers bevægelser vha. video, dvs. automatisk analyse og forståelse af bevægelser udført af personer og optaget af video kameraer. Forskellige former for bevægelser bliver behandlet, fra en- og to-arms bevægelser (som fx at pege, vinke, klappe osv.) til bevægelser med hele kroppen (som fx at gå, løbe, hoppe osv.).

Bevægelserne er optaget med standard farve kameraer, multi-kamera opsætninger og Time-of-Flight (ToF) dybde kameraer, hvilket muliggør analysering af både 2D og 3D data. 2D billede dataen optaget med standard kameraer indeholder både bevægelser udført i simple scener med kontrollerede omgivelser og indstillinger (dvs. én person, fastmonteret kamera, simpel baggrund og ensartet størrelse, rotation, synsvinkel og belysning) og komplekse scener med ukontrollerede omgivelser og indstillinger (dvs. flere personer, friholdt kamera, rodet baggrund og stor variation i størrelse, rotation, synsvinkel og belysning). Til optagelse af 3D data anvendes der både direkte 3D sensorer (ToF dybde kameraer) og 3D rekonstruktion fra flere kameravinkler, for at undersøge hvilken betydning kvaliteten af 3D data har samt fordelen ved hver af teknologierne.

Anvendelse af både globale og lokale billede features undersøges til genkendelse af bevægelser. En metode baseret på globale feature og key-frame princippet præsenteres til at genkende armbevægelser i simple scener. Denne metode finder et sæt af karakteristiske armkonfigurationer og beskriver disse ved hjælp af deres lokale bevægelse. Denne beskrivelse kaldes bevægelsesenheder. En sekvens af bevægelsesenheder kan klassificeres som en armbevægelse ved hjælp af Edit Distance algoritmen i en sandsynlighedsbaseret udgave. Denne 2D metode udvides til en 3D metode ved hjælp af et ToF kamera, der genererer 3D data. De tredimensionelle bevægelsesenheder gør det muligt at genkende armbevægelser uanset kameravinklen.

Til genkendelse af bevægelser udført med hele kroppen i komplekse scener præsenteres en metode, der er baseret på detektion af interesse punkter i tid og rum (STIPs) samt lokalt beskrivelse af billede features. Robuste og selektive STIPs detekteres ved at undertrykke omkringliggende punkter kombineret med lokale og temporale begrænsninger. Metoden er specielt robust overfor kamera bevægelse og rodet baggrund, hvor andre detektorer fejler, og forbedrer ydeevnen ved at detektere flere gentagende, stabile og særprægede STIPs for mennesker, mens uønskede baggrunds STIPs bliver frasorteret.

For at give læseren et overblik over den seneste udvikling inden for genkendelse af bevægelser præsenteres en undersøgelse, som gennemgår og sammenligner nylige metoder til 3D modellering af kroppen, estimering af kropskonfigurationer og genkendelse af bevægelser ved anvendelse af flere kameraer og diskuterer applikationsområder samt de tilhørende krav. For at kunne sammenligne de foreslåede metoder, der ikke kan sammenlignes kvantitativt, udføres en kvalitativ vurdering. Endvidere analyseres anvendeligheden af prominente 3D teknikker til estimering af kropskonfigurationer, hvor ikke blot den udførte bevægelse skal genkendes, men også en mere detaljeret beskrivelse af konfigurationen af kropsdele og led.

Baseret på de fundne resultater af det tidligere arbejde på 2D og 3D genkendelse af bevægelser, udvides idéerne fra detektion af STIP og lokal beskrivelse af billede features til 3D området, for at generere en mere robust og beskrivende 3D repræsentation af bevægelser, som mere effektivt håndterer ændringer i synsvinkel og okklusion. 4D (3D rum + tid) STIPs detekteres i billeder optaget fra flere synsvinkler og en ny lokal beskrivelse af 3D bevægelse, Histogram over Optisk 3D Flow (HOF3D), ekstraheres af estimeret 3D optisk flow i naboområdet af hvert 4D STIP og gøres invariant i forhold til kameravinklen.

Endelig designes og implementeres en metode til automatisk forgrundssegmentering samt detektion af skygger. Forgrundssegmentering er en af de mest anvendte data forbehandlingsteknikker for computer vision algoritmer for at ekstrahere interesse områder, fx til genkendelse af bevægelser, mens indvirkningen af skygger er et velkendt problem i computer vision. Dette arbejde undersøger en flertrins metode til forgrundssegmentering og detektion af skygger. Først præsenteres en bottom-up arkitektur, som anvender en ny teknik baseret på gradient og farve modeller for at kunne separere bevægende objekter fra de kromatiske skygger de kaster. Herefter udvikles en top-down arkitektur baseret på et tracking system, som anvender gensidig information om objekter og skygger for at forbedre detektionen af kromatiske skygger.

Preface

This thesis documents my main research activities from 2006 to 2011. I received my M.Sc.EE in June 2005 and have been working as a research assistant at the Laboratory of Computer Vision and Media Technology, Aalborg University since start 2006.

The work has been funded by different projects giving the thesis different focus points within automatic video-based human action recognition, namely foreground segmentation, shadow detection, tracking and recognition of arm gestures and full-body actions in both 2D and 3D. The thesis consists of a collection of published texts together with an introduction that provides an overview of the topic, reviews the publications, and highlights the main contributions.

The following projects have funded the work presented in this thesis:

BigBrother: Big Brother *is* watching you!, Danish Agency for Science, Technology, and Innovation, 2008-2011

HERMES: Human-Expressive Representations of Motion and Their Evaluation in Sequences, EU project (FP6 IST-027110), 2006-2009

MoPrim: Motion primitives for a communicative human body language, Danish Research Council project, 2004-2007

During my Ph.D. study I have had three external research co-operations with different laboratories located in Europe and USA:

CVRR, UCSD: Ph.D. study abroad at The Computer Vision and Robotic Research Laboratory, University of California, San Diego, collaborative work with Prof. Mohan Trivedi and Cuong Tran on human activity recognition, March–August 2011

AIIA, AUTH: Research stay at The Artificial Intelligence Information Analysis Laboratory, Aristotle University of Thessaloniki, collaborative work with Prof. Ioannis Pitas and Nikolaos Nikolaidis on 3D optical flow estimation for multi-view camera systems. March 2010

CVC, UAB: Research stay at The Computer Vision Center, Universitat Autnoma de Barcelona, collaborative work with Ivan Huerta and Jordi Gonzlez on shadow detection and tracking, January 2010

I have met and cooperated with many highly qualified and inspiring people during the past years who have influenced my work and motivated me. First of all I would like to thank Thomas Moeslund for his guidance throughout this period and for initiating and supporting the process of writing this thesis. I would also like to thank Preben Fihl, Bhaskar Chakraborty, Ivan Huerta, Jordi Gonzlez, Cuong Tran, Mohan Trivedi, Nikolaos Nikolaidis and Ioannis Pitas who have been my closest collaborators at different times.

Michael B. Holte

Aalborg, November 2011

Contents

List of Figures

Chapter 1

Introduction

During the last decade vision-based human activity recognition has been an important topic in the "looking at people" domain [29, 38, 57] of computer vision research. A large number of methods for automatic human activity recognition have been proposed, stretching from human model and trajectory-based methods towards holistic and local feature descriptor-based methods. In recent years a wide range of applications using human activity recognition has been introduced. Among those, several key applications are listed below, including:

Advanced Human-Computer Interaction (HCI) Beyond traditional medium like computer mouse and keyboard, it is desirable to develop better, more natural interfaces between intelligent systems and human in which understanding visual human gesture is an important channel. A few examples are using hand movement to control the presentation slides [24] or recognizing manufacturing steps to help workers to learn and improve their skills [39].

Assisted living Pose estimation and activity recognition can also be applied in assisting handicapped people, elderly people, as well as normal people. For example a system to detect when a person falls [41] or a robot controlled by blinking [5].

Autonomous mental development Study the development of human mental capabilities by observing its real-time interactions with the environment using its own sensors and effectors, e.g. study the cognitive development and learning process of young children [8]. Instead of manually observing the data for analysis, such studies can utilize the recent advances in pose estimation and activity analysis to automate the process and enable analysis in a larger scale.

Gesture-based interactive games In which the player use non-intrusive body movement to interact with the games. For example an Interactive Balloon Game [49] or the well-known Microsoft Kinect Xbox [43].

Intelligent driver assistance systems Looking at driver is a key part required in a holistic approach for intelligent driver assistance systems [50]. Examples of driver assistance systems using posture and behavior analysis are: Monitoring driver awareness based on head pose tracking [33], combining driver head pose and hands track-

ing for distraction alert [48], modeling driver foot behavior to mitigate pedal misapplications [47], developing smart airbag system based on sitting posture analysis [51], or predicting driver turn intent [9].

Movies, 3D TV and animation Human motion capture is also applied extensively in movies, 3D TV and animation. For example in the Avatar movie, in a digital dance lesson [11] or for recording and representation of data for 3D TV [4].

Physical therapy Modern biomechanics and physical therapy applications require the accurate capture of normal and pathological human movement without the artifacts of intrusive marker-based motion capture systems. Therefore marker-less posture estimation and gesture analysis approaches were also developed to be applied in this area [40, 31]

Smart environments In which humans and environment collaborate. Smart environments need to extract and maintain an awareness of a wide range of events and human activities occurring in these spaces [52]. For example, monitoring the focus of attention and interaction of participants in a meeting room [32, 55].

Sport motion analysis Several sports like golf, ballet, or skating require accurate body posture and movement therefore posture estimation and gesture analysis could be applied to this area for analyzing performance and training.

Video surveillance Video surveillance is used in many places such as critical infrastructure, public transportation, office buildings, parking lots, and homes. However manually monitoring these cameras is becoming a hazard. Therefore approaches for automatic video surveillance including outdoor human activity analysis, e.g. [35, 54] will be needed.

Video annotation With the development of hardware technology, a very large amount of video data can be easily saved. Among those, there are lots of human related videos such as surveillance videos, sport videos, or movies. Instead of manually scanning through those large video database to get the needed information, human motion analysis can be used to annotate those video, e.g. approaches to annotate video of a soccer game [6] or in more general for outdoor sports broadcasts [22].

For some applications the videos are captured specifically with human motion analysis and activity recognition in mind. *E.g.*, autonomous mental development, physical therapy, and sport motion analysis, which allows physicians and coaches to conduct a much more thorough analysis. The resolution is usually high, the background is clean and simple (blue screen) or synchronized video from multiple viewpoints can be recorded, enabling easy and high quality segmentation of the object of interest. The same goes for advanced human-computer interaction, gesture-based interactive games, movies, 3D TV and animation, however, the surroundings and the way the activities are performed are rarely controllable, leading to a more difficult analysis and recognition. For surveillance applications, video annotation, smart environments, assisted living and intelligent driver assistance systems the video is most often captured in unconstrained environments with people doing real-life motions rather than instructed performances of specific actions. Such unconstrained

video makes the recognition of human activities a very challenging task. Additionally, real-time performance is another important issue for several applications.

The computer vision research presented in this thesis does not target a specific application of human activity recognition but rather presents work on systems that can enable many different applications.

The focus of this thesis will be specified next (section 1.1), and hereafter the remainder of this chapter will be structured as follows. Section 1.2 will elaborate on the contents of the thesis by presenting an outline of each of the chapters and state how the different methods and chapters relate to each other. In section 1.3 the main contributions of this thesis will be highlighted. Section 1.4 gives a comparative dataset listing and section 1.5 lists all the publications that have been published in relation to the work of this thesis.

1.1 The focus of this thesis

The work described in this thesis deals with video-based human activity recognition, *i.e.* automatic analysis and understanding of activities performed by people and recorded by video. Different kind of activities are considered from one and two arms gestures (*e.g.* point, wave, clap etc.) to full-body actions (*e.g.* walk, run, jump etc.). The activities are recorded by standard color cameras, multi-view camera setups and time-of-flight (ToF) range cameras, enabling analysis of both 2D and 3D video data. The 2D image data recorded by standard color cameras captures both activities performed in simple scenes with controlled settings (*i.e.* one actor, steady camera, simple and clean background, and low variation in scale, rotation, viewpoint and illumination) and complex scenes with unconstrained settings (*i.e.* multiple actors, moving camera, background clutter, and high variation in scale, rotation, viewpoint and illumination).

Activity recognition in laboratory environments or carefully designed scenes can provide valuable information about the performance of methods and systems. However, the variability and challenges of real-life scenes are not investigated in this way, often leading to less general and less applicable methods. Hence, the work of this thesis also focuses on scenes that are not carefully constrained. The work on 2D action recognition specifically targets dynamic outdoor scenes and addresses multiple people. One part of the work on 2D activity recognition is applied in an office environment, whereas the other part addresses the challenges of dynamic outdoor scenes.

For acquisition of 3D data both direct 3D imaging devices (ToF range cameras) and 3D reconstruction from multiple camera views are applied, to explore challenges of different quality of 3D data and the advantages of each technology. In contrast to 2D activity recognition, 3D approaches are more confined towards indoor scenes, due to the nature of the current sensors. ToF range cameras are usually limited to a range up to about 6-7 meters, while 3D reconstruction from multiple camera views are limited to the overlapping area of the camera views. Hence, the work on 3D activity recognition presented in this thesis focus solely on indoor scenes.

Human activity recognition is a very active field of research which for example can be seen by the number of publications reviewed in recent surveys [3, 19, 20, 29, 38, 57].

The amount of publications within human activity recognition results in surveys and reviews that focus on specific areas, *e.g.* [3] focus on body modeling and recognition of actions and interactions, [19, 20] present a review focusing purely on view-invariant pose estimation and action recognition, [38] reviews solely 2D action recognition, and [57] both 2D and 3D action recognition. While [29] gives a more broader review of human motion capture, dealing with tracking, pose estimation and recognition. To give the reader an overview of recent developments in human activity recognition a survey is presented in this thesis. The survey reviews and compare recent approaches for multi-view human 3D body modeling, pose estimation and human activity recognition. Human body modeling and pose estimation are a important topics to consider, when dealing with human activity recognition, since some applications requires estimation of exact body pose and positions of joints and body parts, *e.g.* HCI, interactive games, physical therapy, sport motion analysis, movies, 3D TV and animation. Hence, this survey reviews recent work in all three topics using multi-view videos.

Finally, an approach for automatic foreground segmentation and shadow detection is designed and implemented. Foreground segmentation is one of the most used preprocessing steps for many computer vision algorithms to extract regions of interest. Although the methods for human activity recognition described in this thesis do not apply complex foreground segmentation (the shape-from-silhouette technique for 3D reconstruction from multi-view video use foreground segmentation achieved by simple background subtraction), this preprocessing step is very important and crucial for the performance of any methods for tracking and recognition. Foreground segmentation in long video sequences of complex and unconstrained scenes is a challenging task, due to diverse type of scenes, background clutter, camera motion, occluded bodies, unconstrained human motion, interaction, grouping and challenging lighting conditions and illumination variations. To this end, the impact of shadows is a notorious problem in computer vision. Shadows can take any size and shape, can exhibit different chromaticity than the background, and their intensity values can be similar to those of any new object appearing in a scene. Consequently, shadows can be very difficult to detect, and therefore usually detected as a part of moving objects. The impact of shadows can be crucial for the foreground segmentation, and cause objects to merge, distort their size, shape, and appearance. This results in a reduction of computer vision algorithms' applicability for, *e.g.*, scene monitoring, object and activity recognition, target tracking and counting. Although, detection and removal of shadows is important for successful and precise foreground segmentation, the problem of shadow detection is still far from being solved.

1.2 Overview of this thesis

This thesis consists of eight chapters with the current chapter being the first. The following six chapters each consists of a previously published text. Each chapter has a brief introduction explaining the context of the publication. The eighth chapter concludes on the thesis. The following will give an overview of each of the chapters, presenting and outline of the used methods and a summary of the results.

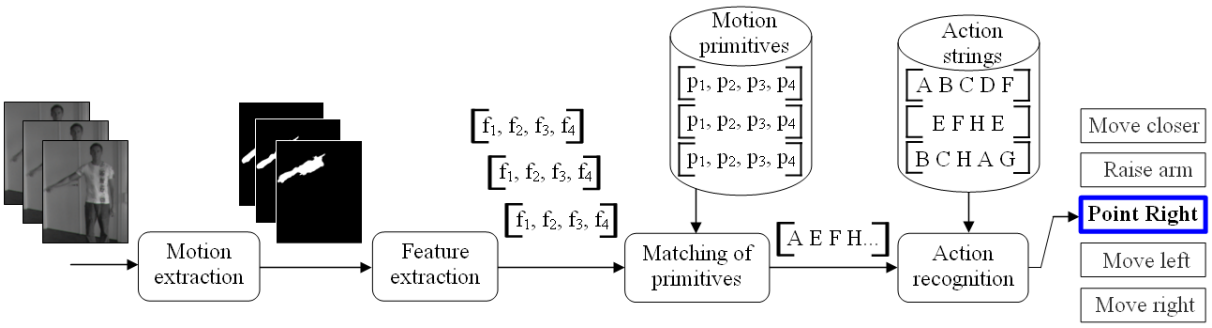


Figure 1.1: An overview of the approach for recognition of human gestures. Motion is extracted and represented with a four-dimensional feature vector. The feature vectors are matched to a set of trained motion primitives. This results in a sequence of primitives representing the gesture performed in the input video. The gesture recognition step compares an incoming sequence of primitives with a set of trained gesture models and classifies the gesture.

Chapter 2. 2D Human Gesture Recognition

This first chapter on gesture recognition investigates a key-frame approach for the recognition of arm gestures. The gestures are five one-arm gestures used in the communication between people over longer distances. The gestures are: point right, move right, move left, move closer, and raise arm.

Many recent methods recognize gestures or actions directly on image data rather than extracting joint locations and then recognize gestures based on this representation. Gesture recognition from silhouette data is an example of this. Another very popular approach in recent years has been recognition using spatio-temporal interest points or spatio-temporal volumes that both base the recognition on information from all frames of a sequence, which is also the approach taken to recognize actions in chapter 4. In chapter 2 an approach is presented that base the gesture recognition on a small set of characteristic poses (key-frames) that can be reliably detected. Key-frame approaches base the recognition on a subset of the entire sequence with the assumption that certain characteristics of a gesture are more easily recognized than others, and basing the recognition on the frames containing these characteristics yields a more robust result. Recent and extensive surveys on action recognition are given in [38, 57, 19].

The approach of chapter 2 finds the characteristic arm poses based on the arm motion for each pose. The representation used for these poses are denoted *m*otion primitives. Figure 1.1 gives an overview of the approach.

Motion is detected using double differencing, *i.e.* using three consecutive frames to generate two difference images which are thresholded and combined by a pixel-wise logical AND. The thresholding utilizes a hysteresis principle to eliminate noise. The result of the motion detection is a binary blob describing the motion of the arm.

To extract a set of features for the motion blob it is modeled by an ellipse and four scale invariant features are calculated. The features describe the shape, orientation, and location of the ellipse with the location defined relative to a reference point on the person.

Based on the extracted feature vector each incoming frame will be classified as either belonging to one of the motion primitives or as a noise class.

The representation of the motion primitives is based on a set of training samples for each pose. To acquire the training data magnetic trackers are placed at the joints of the arm on training subjects. Each training subject repeats all arm gestures and the trajectories of the tracker markers are transferred to a computer graphics model of a person. The animations of the graphics model constitute the training data for the approach. This semi-synthetic training data (*i.e.* gestures performed by humans but synthesized with a graphics model) decouples the training data from the image data used in the recognition process. For other approaches, like methods based on spatio-temporal interest points, the training and testing data are typically different but with the exact same image characteristics. The motion-based primitives and the semi-synthetic training data make the approach presented in chapter 2 applicable to more diverse input data.

The subsequences defining the motion primitives (three frames for each primitive) are found manually. The criteria for selection of the subsequences are the following: Firstly, that the subsequence represents a characteristic and representative 3D configuration. Secondly, that a certain amount of motion is present in the subsequence. Thirdly, that the subsequence is representative for as many gestures as possible. The third criteria results in a small set of 10 robust primitives for the five gestures with each gesture represented by five to eight of these primitives.

The set of semi-synthetic subsequences that represent the training data for a motion primitive is processed to find the feature vectors and each primitive is represented by the mean and covariance for these vectors. The feature vectors of testing video can now be classified using the Mahalanobis distance. If the minimum Mahalanobis distance is above a certain threshold then the feature vector is classified as noise. For a test video this classification will result in a sequence of primitives representing the gesture being performed in the video.

The classification of a sequence of primitives as one of the five gestures is done using a novel extension of the edit distance that incorporates the likelihood of each primitive. The original edit distance expresses the number of operations needed to convert one sequence into another where possible operations are insertion of a symbol, deletion of a symbol, or exchange of a symbol with one from the other sequence. Each operation can have an associated cost which in chapter 2 is extended to a cost dependent on the probability of each observed symbol, here being each motion primitive. The probability of a motion primitive is expressed through the number of observations of a given primitive. The edit distance is furthermore normalized with the length of the sequence of primitives to avoid bias towards short sequences.

To test the method a set of 550 video sequences is captured, each containing the execution of a gesture. Two different test setups are used. In the first setup each test video contains the execution of one gesture. This could imitate gesturing for human-computer interaction where it is known when commands are being issued. The second test setup imitates the more realistic problem of not knowing when the execution of the gesture commences and when it terminates. This is achieved by adding executions of half of a gesture to the start and end of the original captured video sequences. The gestures used for these half

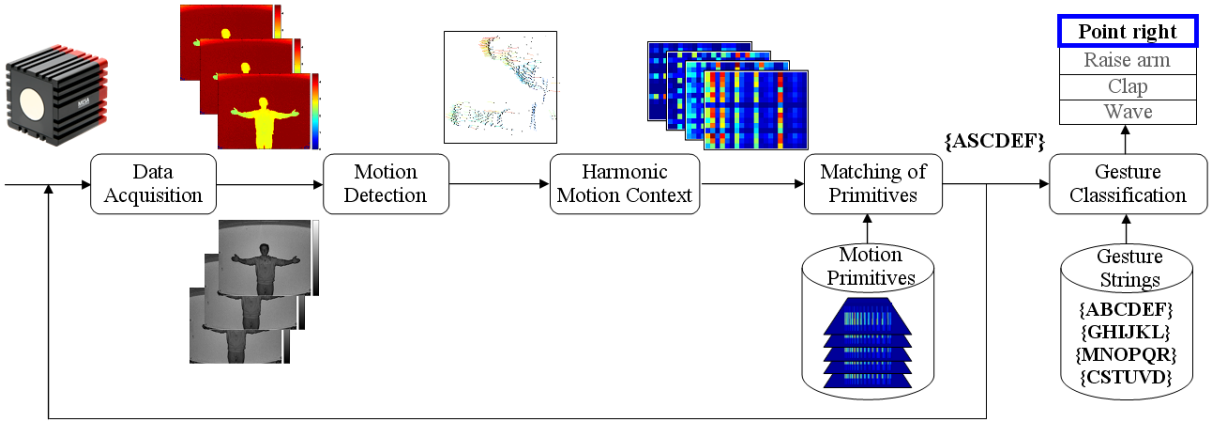


Figure 1.2: A schematic overview of the 3D human gesture recognition. Recognition is based on range and intensity images. 3D motion is extracted and described by harmonic motion contexts. These are matched to trained motion primitives. A number of frames are processed this way (illustrated by the feed back loop) resulting in a sequence of primitives. This sequence is classified against a set a trained gesture strings.

executions are chosen randomly. The overall recognition rates for the two test setups respectively are 88.7% and 85.5%. Most of the erroneous classifications are a result of confusion between the *m*ove closer and *r*aise arm gestures. Seen from a fronto-parallel view the motion of the two gestures are somewhat alike. This issue is one of the motivating factors for extending the work of chapter 2 into a view-invariant representation which is presented in chapter 3.

Chapter 3. 3D Human Gesture Recognition

The approach of chapter 2 is in this chapter extended into a view-invariant gesture recognition method based on 3D input data. A time-of-flight range camera is used to produce both a depth map and an intensity image which allow for the extraction of motion in 3D. The time-of-flight camera ensures a direct alignment of the depth and intensity information as opposed to the classical stereo approaches which have to be carefully calibrated and establish correspondences between cameras.

Another important difference between the approach presented here and other related methods is the characteristics of the training data. When addressing invariance to viewpoint in gesture recognition the training data often includes video captured from different viewpoints. A view-invariant representation of the gestures ensures that a test video can be classified without first recovering the viewpoint of that sequence (see for example [45] and [56]). The approach presented in chapter 3 reduces the training data to a single viewpoint while maintaining the ability to recognize gestures from different viewpoints, say viewpoints rotated ± 45 degrees from the training viewpoint.

The recognition is based on the notion of motion primitives. Here, the motion primitives describe both the amount of motion and the 3D direction of the motion (as opposed to the binary motion detection of chapter 2). Figure 1.2 shows an overview of the method. The

motion is detected using a 3D version of optical flow. The detected motion is represented using motion contexts (an extension of Shape Contexts). The representation is made invariant to rotation around the vertical axis using spherical harmonic basis functions, yielding a harmonic motion context representation.

In each frame the motion primitive which best explains the observed data is found. This is done by calculating the normalized correlation coefficients between the harmonic motion contexts of the observed data and the motion primitives. A video sequence will in this way result in a sequence of primitives representing the gesture performed in the video. The classification of a sequence of primitives is done by use of the probabilistic edit distance of chapter 2.

The method is used to recognize four one- and two arms gestures, namely "point right", "raise arm", "clap", and "wave". These four gestures are represented using 22 motion primitives. The method is tested on 160 video sequences. The sequences show 10 test subjects performing two repetitions of each gesture. The gestures are captured from two viewpoints, one frontal view and one view rotated 45 degrees. As stated above, only data from the frontal view is used for training while the testing includes both viewpoints. The test uses the same test protocol as in the 2D gesture recognition system, where one test is conducted with exactly one gesture per sequence (known start and end times) and one test has "noise" gestures (half executions of gestures) added to the beginning and end of the sequences (unknown start and end times). The method achieves a recognition rate of 94.4% when the start and end times are known and a recognition rate of 86.9% when the start and end times are unknown.

Chapter 4. 2D Human Action Recognition

Chapter 4 presents an approach for action recognition based on detection of spatio-temporal interest points (STIPs) and local description of image features. STIPs was first introduced by Laptev et al. [23] and has become very popular for human action recognition. Later a number of other methods for STIP detection have been proposed by other authors [10, 18, 34, 59, 60]. In common for these STIP detectors is that they are sensitive to camera motion and background clutter, resulting in a larger amount of detected interest points in the background.

To counter this problem chapter 4 propose a a novel approach for robust and selective STIP detection, by applying surround suppression combined with local and temporal constraints. The method separates space and time by first detecting Spatial Interest Points (SIPs), then suppressing unwanted background points, and finally imposing local and temporal constraints, achieving a set of selective STIPs which are more robust to these challenges. This new method is significantly different from existing STIP detection techniques, which detect interest points directly in a spatio-temporal space, and improves the performance by detecting more repeatable, stable and distinctive STIPs for human actors, while suppressing unwanted background STIPs.

For action representation a bag-of-video words (BoV) model of local N-jet features is applied to build a vocabulary of visual-words. BoV models have become very popular for representing a large amount of different image features and has successfully been applied

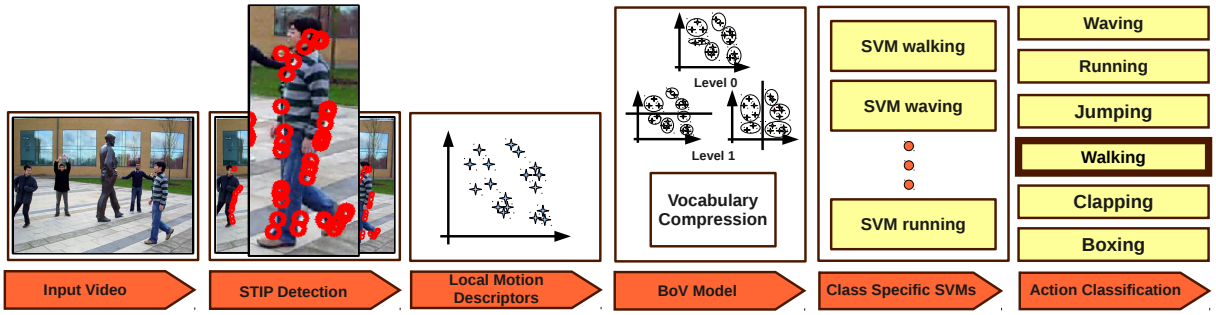


Figure 1.3: A schematic overview of the system structure and data flow pipeline of the 2D human action recognition. Spatio-temporal interest points are detected in the input video, and local N-jet features are extracted at the specific locations. A bag-of-video words vocabulary is build using spatial pyramids and compressed into a final set of video-words. Finally, class specific support vector machines classifiers are trained and used for classification of actions.

by numerous authors [10, 23, 25, 26, 60, 61]. To this end, a novel vocabulary building strategy is introduced by combining spatial pyramid and vocabulary compression techniques, resulting in improved performance and efficiency. Action class specific Support Vector Machine (SVM) classifiers are trained for categorization of human actions. Different SVM kernels are tested, where the χ-square achieves the best performance on the Weizmann dataset. Figure 1.3 gives a schematic overview of the proposed approach.

The selective STIP detector is evaluated separately on MSR I and Multi-KTH by estimating a score for the number of detected STIPs for the actors in comparison to those detected in the background. This STIP detection ratios: the number of STIPs detected on the actors with respect to the total number of detected STIPs is compared to other STIP detectors, and the selective STIP detector shows superior performance on 76.21% for MSR I and 90.34% for Multi-KTH.

A comprehensive set of experiments on popular benchmark datasets (KTH and Weizmann), more challenging datasets of complex scenes with background clutter and camera motion (CVC and CMU), movie and YouTube video clips (Hollywood 2 and YouTube), and complex scenes with multiple actors (MSR I and Multi-KTH), validates the approach and show state-of-the-art performance. For Weizmann the recognition accuracy is 99.50%; KTH 96.35%; CVC 100%; CMU 99.42%; Hollywood 2 58.45%; and YouTube 86.98%.

Additionally, cross-data action recognition is reported by training on source datasets (KTH and Weizmann) and testing on completely different and more challenging target datasets with shared actions (CVC, CMU, MSR I and Multi-KTH). For this cross-data evaluation the recognition accuracy for Weizmann is 100%; CVC 96.95%; CMU 91.94%; MSR I 84.77%; and Multi-HTH 98.40%. This documents the robustness of the proposed approach in the realistic scenario, using separate training and test datasets, which in general has been a shortcoming in the performance evaluation of human action recognition techniques.

Due to the unavailability of ground truth action annotation data for the Multi-KTH dataset, an actor specific spatio-temporal clustering of STIPs is introduced to address the problem of automatic action annotation of multiple simultaneous actors. When this

Figure 1.4: The application domain of human 3D body modeling, pose estimation and activity recognition, covering: advanced Human-Computer Interaction (HCI), assisted living, gesture-based interactive games, intelligent driver assistance systems, movies, 3D TV and animation, physical therapy, autonomous mental development, smart environments, sport motion analysis, video surveillance and Video annotation.

spatio-temporal clustering technique is used instead of ground truth bounding boxes for Multi-KTH the recognition accuracy is 94.20%.

Chapter 5. A Survey on Multi-View Human 3D Body Modeling, Pose Estimation and Activity Recognition

In chapter 5 a survey on human 3D body modeling, pose estimation and activity recognition from multi-view videos is presented. This survey gives an overview and comparative study of recent developments. The survey is build upon the application domain of human 3D body modeling, pose estimation and activity recognition, covering: advanced Human-Computer Interaction (HCI), assisted living, gesture-based interactive games, intelligent driver assistance systems, movies, 3D TV and animation, physical therapy, autonomous mental development, smart environments, sport motion analysis, video surveillance and Video annotation. Figure 1.4 gives illustrative examples of these applications.

The requirements of the different applications are analyzed, revealing that the requirements vary significantly depending on the desired application. This results in the need of

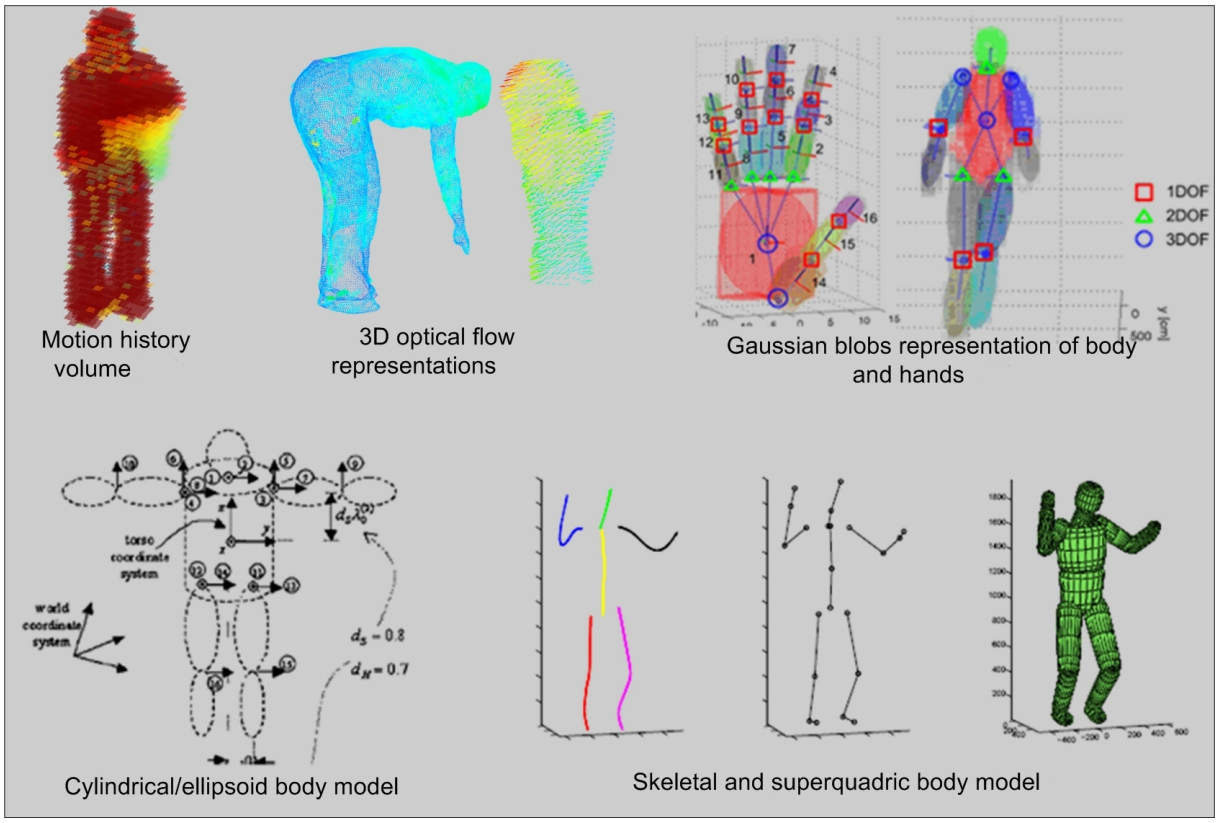


Figure 1.5: Prominent 3D human body model and human motion representations: motion history volumes, 3D optical flow, Gaussian representation with different levels of detail, cylindrical/ellipsoid, skeletal and superquadric body model.

approaches, which e.g. can operate on different abstraction levels, in uncontrolled environments, with high precision, in critical real-time and for large database search. Based on the application domain and the associated requirements, chapter 5 gives a detailed description and comparison of some prominent and diverse 3D pose estimation techniques, which represent the contributions to this field well. Furthermore, a quantitative comparison of several promising multi-view human action recognition approaches is presented, covering both 2D and 3D multi-view methods, using two publicly available datasets: the INRIA Xmas Motion Acquisition Sequences (IXMAS) Multi-View Human Action Dataset [56] and the i3DPost Multi-View Human Action and Interaction Dataset [12]. The qualitative comparison reveals that methods using 3D representations of the data turn out to outperform the 2D methods. Some of the most prominent 3D human body model and human motion representations are shown in figure 1.5.

Multi-view camera systems have the advantage that they enable full 3D reconstruction of the human body, and to some extent handles self-occlusion. In contrast single 3D imaging devices, like ToF sensors and Kinect, will only acquire 3D surface structure visible from that single viewpoint. Although the reviewed approaches show promising results for multi-view human body modeling, pose estimation and action recognition, 3D reconstructed data from multi-view camera systems has some shortcomings. First of all, the quality of the silhouettes is crucial for the outcome of applying Shape-from- Silhouettes. Hence,

shadows, holes and other errors due to inaccurate foreground segmentation will affect the final quality of the reconstructed 3D data. Secondly, the number of views and the image resolution will influent the level of details which can be achieved, and self-occlusion is a known problem when reconstructing 3D data from multi-view image data, resulting in merging body parts. Finally, 3D data can only be reconstructed in a limited space where multiple camera views overlap.

Chapter 6. Multi-View Human Action Recognition

Chapter 4 deals with human action recognition using a single camera. Monocular methods will however always be challenged by larger changes in viewpoint and heavy occlusions. Chapter 6 introduces a multi-view approach that extends the work of chapter 3 and 4 to generate a more robust and descriptive 3D representation of human actions, which more efficiently deals with the problems of viewpoint changes and occlusion.

A 3D data representation is more informative than the analysis of 2D activities carried out in the image plane, which is only a projection of the actual actions. As a result, the projection of the actions will depend on the viewpoint, and not contain full information about the performed activities. To overcome this shortcoming the use of 3D data has been introduced through the use of two or more cameras. In this way the surface structure or a 3D volume of the person can be reconstructed, e.g., by Shape-From-Silhouette (SFS) techniques [46], and thereby a more descriptive representation for action recognition can be established.

2D human action recognition has moved from model-based approaches to model-free approaches using local motion features. In this context, methods based on Spatio-Temporal Interest Points (STIPs) and Bag-of-Words (BoW) are successfully applied to this area. On the contrary, 3D Human action recognition is more confined towards model-based approaches or holistic features. To minimize this gap, chapter 6 contribute to the field of multi-view human action recognition, by introducing a novel 3D action recognition approach based on detection of 4D (3D space + time) Spatio-Temporal Interest Points and local description of 3D motion features extracted from reconstructed 3D data acquired by multi-camera systems. Opposed to other methods for 3D action recognition, which are solely based on holistic features, e.g. [15, 36, 45, 56], the presented approach extends the concepts of STIP detection and local feature description for building a Bag-of-Words (BoW) vocabulary of human actions, which has gained popularity in the 2D image domain, to the 3D case. Figure 1.6 show a schematic overview of the multi-view approach.

STIPs are detected in multi-view images in a selective manner by surround suppression of the output of the basic Harris corner detector and imposing local spatio-temporal constraints, as described in chapter 4. Hereafter, the multi-view image STIPs are extended to 4D using 3D reconstructions of the actors and pixel-to-vertex correspondences of the multi-camera setup. By introducing a novel local 3D motion descriptor, called Histogram of Optical 3D Flow (HOF3D), estimated 3D optical flow is represented in the neighborhood of each 4D STIP, and four solutions to make the HOF3D descriptor view-invariant are examined: (i) vertical rotation with respect to the orientation of the normal vector and (ii) the orientation of the velocity vector, (ii) circular bin shifting with respect to the

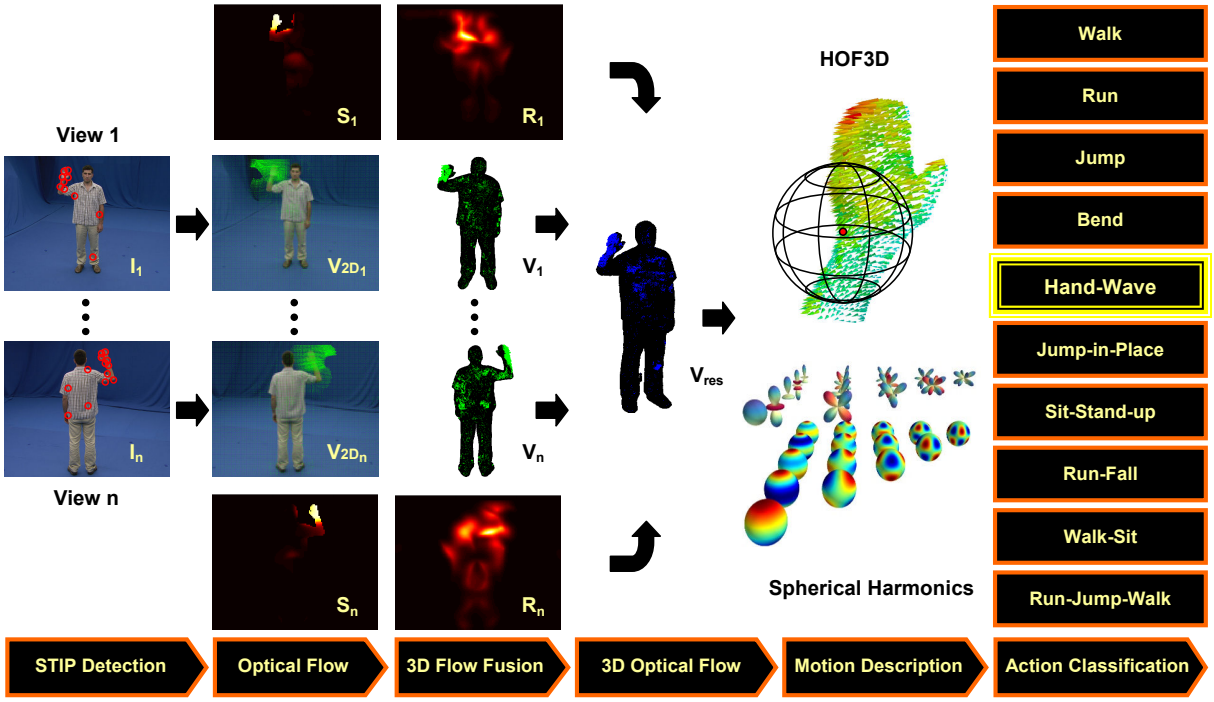


Figure 1.6: A schematic overview of the multi-view human action recognition. Spatio-Temporal Interest Points (STIPs) are detected in multi-view images in a selective manner and optical flow is estimated for each view. Hereafter, the multi-view image STIPs are extended to 4D (3D space + time) and the optical flow to 3D using 3D reconstructions of the actors and pixel-to-vertex correspondences of the multi-camera setup. The estimated 3D optical flow is represented in the neighborhood of each 4D STIP using a novel local 3D motion descriptor, called Histogram of Optical 3D Flow (HOF3D), which is made view-invariant, *e.g.* by decomposing the representation into a set of spherical harmonic basis functions. Actions are recognized by building a Bag-of-Words (BoW) vocabulary of view-invariant HOF3D descriptors, which is organized in 3D spatial pyramids, and further compressed and classified using Agglomerative Information Bottleneck (AIB) and Support Vector Machines (SVM), respectively.

horizontal mode of the histogram and (iv) by decomposing the representation into a set of spherical harmonic basis functions. The local HOF3D descriptors are divided using 3D spatial pyramids to capture and improve the discrimination between arm- and leg-based actions. Additionally, two pyramid divisions based on a horizontal plane estimated as (i) the center of gravity of the 3D human model and (ii) the center of gravity of the detected STIPs. Based on these pyramids of HOF3D descriptors a Bag-of-Words (BoW) vocabulary of human actions is build, which is compressed and classified using Agglomerative Information Bottleneck (AIB) and Support Vector Machines (SVM), respectively.

The approach is evaluated by conducting experiments reported on the publicly available i3DPost [12] and IXMAS [56] datasets, and show promising state-of-the-art results: 98.44% for i3DPost and 100% for IXMAS. Furthermore, an incremental analysis is presented investigating the performance boost of applying 3D spatial pyramids (∼5.5%) and

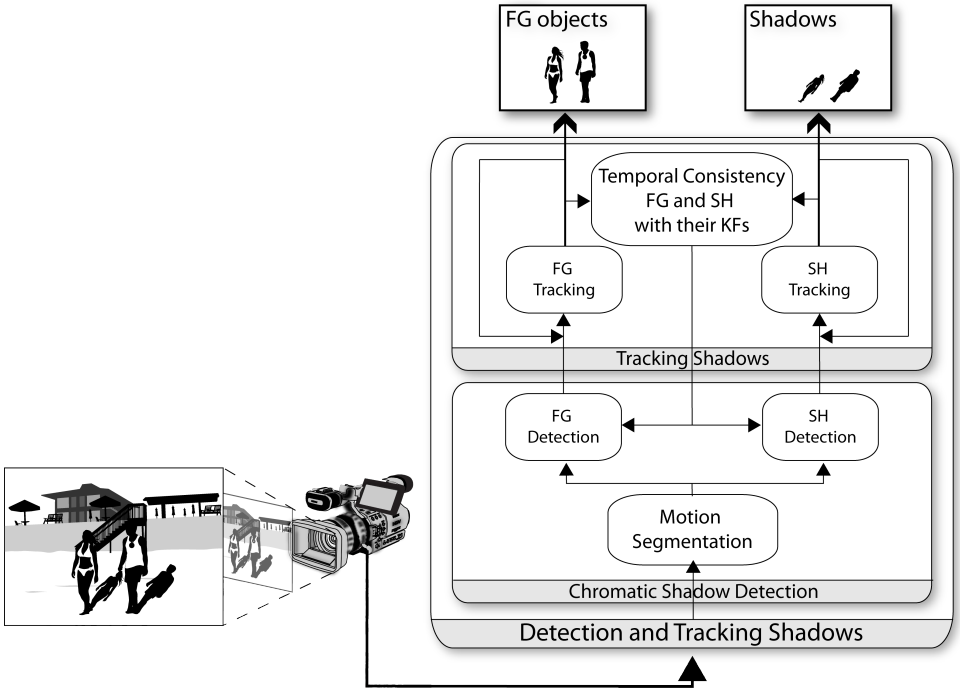


Figure 1.7: A schematic overview of the foreground segmentation and shadow detection. The approach consist of two main parts: a bottom-up chromatic shadow detection module and a top-down shadow tracking module.

vocabulary compression (∼3%). Finally, the view-invariance of the approach is evaluated by training and testing on all different combination of the 8 cameras used to produce the i3DPost dataset, showing the recognition accuracy is quite stable over all view combinations (∼91% ± 6%).

Chapter 7. Foreground Segmentation and Shadow Detection

A fundamental problem for all automatic video surveillance systems is to detect objects of interest in a given scene. A commonly used technique for segmentation of moving objects is background subtraction [29]. This involves detection of moving regions (i.e., the foreground) by differencing the current image and a reference background image in a pixel-by-pixel manner. Usually, the background image is represented by a statistical background model, which is initialized over some time period. An important challenge for foreground segmentation is the impact of shadows. Shadows can be divided into two categories: *static shadows* and *dynamic (moving) shadows*. Static shadows occur due to static background objects (e.g., trees, buildings, parked cars, etc.) blocking the illumination from a light source. Static shadows can be incorporated into the background model, while dynamic shadows have shown to be more problematic. Dynamic shadows are due to moving objects (e.g., people, vehicles, etc.). The impact of dynamic shadows can be crucial for the foreground segmentation, and cause objects to merge, distort their size and shape, or occlude other objects. This results in a reduction of computer vision algorithms' applicability for, e.g, scene monitoring, object recognition, target tracking

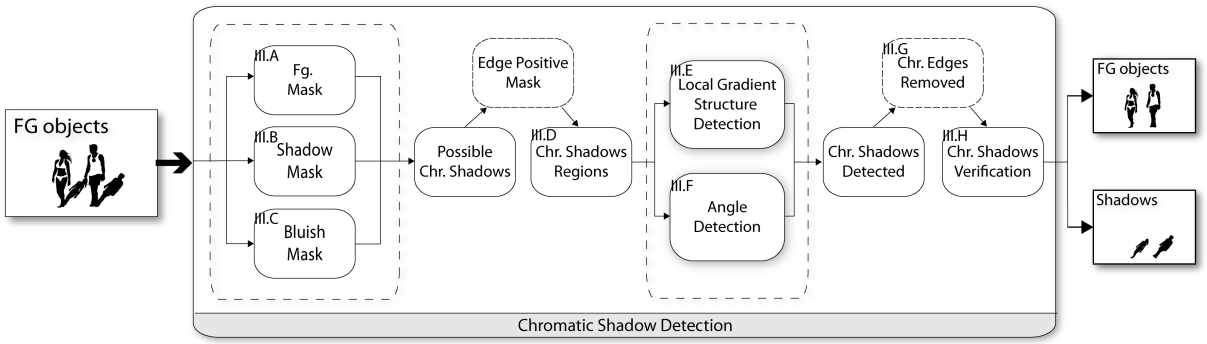


Figure 1.8: A schematic overview of the bottom-up chromatic shadow detection. Gradient and color models are used to separate chromatic moving shadows from detected moving objects. Next, regions corresponding to potential shadows are grouped by considering the " bluish effect" and an edge partitioning. Lastly, temporal similarities between local gradient structures and spatial similarities between chrominance angle and brightness distortions are analyzed for all potential shadow regions, in order to finally identify umbra shadows

and counting.

Dynamic shadows can take any size and shape, and can be both *umbra* (dark shadow) and *penumbra* (soft shadow) shadows. Penumbra shadows exhibit low values of intensity but similar chromaticity values w.r.t. the background, while umbra shadows can exhibit different chromaticity than the background, and their intensity values can be similar to those of any new object appearing in a scene. When the chromaticity of umbra shadows differs from the chromaticity of the global background illumination, we define this as *chromatic shadow*. Consequently, umbra shadows are significantly more difficult to detect, and therefore usually detected as a part of moving objects. When a shadow has successfully been detected it is usually removed instantly, since it is the object which is of interest for further processing and not the shadow. As a result, the information the shadow brings is lost. An interesting idea is to use this information to improve other aspects of object and shadow detection and tracking. Concretely, if a detected shadow is tracked over time instead of being discarded, it could be used to improve the shadow detection and possible the object detection and tracking as well.

In chapter 7, firstly a bottom-up approach for detection and removal of chromatic moving shadows in surveillance scenarios is presented [16]. Secondly, a top-down approach based on a tracking system is proposed in order to enhance the chromatic shadow detection. Figure 1.7 shows a schematic overview of the entire system.

The bottom-up part consists of a novel technique using gradient and color models for separating chromatic moving shadows from detected moving objects. A chromatic invariant color cone model and an invariant gradient model are built to perform automatic segmentation while detecting potential shadows. Next, regions corresponding to potential shadows are grouped by considering "a bluish effect" and an edge partitioning. Lastly, temporal similarities between local gradient structures and spatial similarities between chrominance angle and brightness distortions are analyzed for all potential shadow regions, in order to finally identify umbra shadows. A schematic overview of the bottom-up

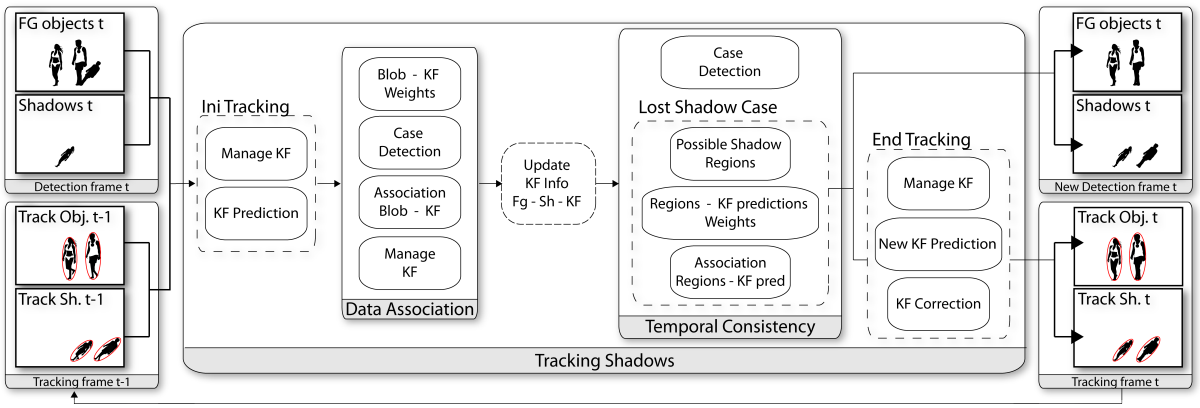


Figure 1.9: A schematic overview of the top-down tracking of foreground objects and shadows to enhance the chromatic shadow detection. A Kalman filter (KF) is created and associated to each foreground object and shadow. Next, an event analysis of the different data association cases are performed. Based on this event analysis, a test for temporal consistency in the association between foreground (FG), shadow (SH) and their respective Kalman Filters is performed. Finally, the tracking results are used as feedback to improve the object and shadow detection, by recovering miss-detected shadows.

part is given in Figure 1.8.

In the top-down part, after detection of objects and shadows, both are tracked using Kalman filters, in order to enhance the chromatic shadow detection. This implies data association between the blobs (foreground and shadows) and Kalman filters using Probabilistic Appearance Models. Next, an event analysis of the different data association cases are performed. Based on this event analysis, we test for temporal consistency in the association between foreground (FG), shadow (SH) and their respective Kalman Filters. Finally, the tracking results are used as feedback to improve the object and shadow detection, by recovering miss-detected shadows. A schematic overview of the top-down part is given in Figure 1.9.

Chapter 8. Conclusion

The conclusion of the thesis will summarize and discuss the main contributions and provide an overview of future research related to human activity recognition.

1.3 Contributions

This section will point out the main contributions presented in this thesis.

Motion primitives for activity recognition In chapter 2 the simple principle of image differencing is used to extract motion primitives which are represented with a compact four-dimensional feature vector. These simple and compact motion primitives achieves good recognition results on arm gestures but are not limited to this

type of activities. The motion primitives are a principled representation that can be used for a wide variety of activities. The gesture recognition by probabilistic edit distance can also easily be used to classify other actions that are represented by a set of primitives. In chapter 3 the approach is also used to recognize gestures in 3D data captured by a time-of-flight sensor.

Synthetic training data for activity recognition The use of a computer graphics model of a person for generation of training data in action recognition methods is explored in chapter 2. Real human motion is captured using a motion capture system and then synthesized using the computer graphics model. The synthetic data provides sufficient training data to obtain good recognition results and at the same time decouples the training data from the test data.

Selective spatio-temporal interest points Current spatio-temporal interest point (STIP) detectors are vulnerable to camera motion and background clutter, and therefore detects large amounts of STIPs in the background in video of complex scenes where these effects appear. In chapter 4 a selective STIP detector is introduced, which detects interest points by separating time and space to suppress background spatial interest points (SIPs) and hereafter impose local and temporal constraints, resulting in more robust STIPs for actors and less unwanted background STIPs. The strong aspect of our proposed STIP detection method is, it can detect dense STIPs at the motion region without being affected by the complex background. This is an important property to detect actions in complex scenarios. To represent actions from local features extracted at each STIP, a novel vocabulary building strategy is proposed by combining spatial pyramids and vocabulary compression. State-of-the-art performance has been achieved using this strategy.

A survey on multi-view approaches In chapter 5 we present a review and comparative study of recent developments in human 3D body modeling, pose estimation and activity recognition. This survey is different from other surveys [19, 28, 29, 37, 38, 57, 58], in the sense that it focus exclusively on recent work on multi-view human body modeling, pose estimation and action recognition, both based on 2D multi-view data and reconstructed 3D data, acquired with standard cameras. The review is organized with respect to the application domain and the associated requirements, and a qualitative and quantitative comparison of the methods are presented to inform the reader on advantages and disadvantages of different approaches. The quantitative comparison of recent approaches for human action recognition reveals that methods using 3D data in general seems to outperform methods using 2D video acquired by multiple cameras. Finally, some of the shortcomings of multi-view camera setups are discussed and thoughts on future directions of 3D body pose estimation and human action recognition are outlined to inspire new research in this field.

3D Optical flow for activity recognition chapter 3 and 6 describe how 3D optical flow efficiently can be estimated for both 3D data acquired by range cameras and multi-camera setups, respectively. First, Motion is detected by computing optical flow in the 2D video, and hereafter extended to 3D optical flow using the depth information acquired from only one viewpoint by a range camera, or by estimating pixel-to-vertex correspondences for reconstructed 3D data. In case of multiple

views, the resulting 3D optical flow for each view is combined into 3D motion vector fields by taking the significance of local motion and its reliability into account. In comparison to other methods for 3D motion detection, *e.g.* motion history volumes [56], 3D optical flow gives detailed 3D motion information represented by both the amount and direction of the motion.

View-invariant 3D motion description In chapter 3 the estimated 3D optical flow for human actors is efficiently represented by their *motion context*. The motion context is an extended version of the regular shape context [7], and represents 3D optical flow by using both the location of motion, together with the amount of motion and its direction. The motion descriptor is made invariant to rotation around the vertical axis by re-representing the motion context using *spherical harmonic basis functions*, yielding a *harmonic motion context* representation. In chapter 6 a local version of the descriptor is proposed, denoted *Histogram of 3D Optical Flow* (HOG3D).

A local descriptor-based strategy for 3D activity recognition The survey presented in chapter 5 reveals how 3D Human action recognition is confined towards model-based approaches or holistic features. To minimize this gap, chapter 6 contribute to the field of 3D human action recognition, by introducing a novel 3D action recognition approach based on detection of 4D Spatio-Temporal Interest Points and local description of 3D motion features extracted from reconstructed 3D data acquired by multi-camera systems. Opposed to other methods for 3D action recognition, which are solely based on holistic features, e.g. [15, 36, 45, 56], the approach extends the concepts of STIP detection and local feature description for building a Bag-of-Words (BoW) vocabulary of human actions presented in chapter 4, which has gained popularity in the 2D image domain, to the 3D case. Results for two publicly available datasets shows promising state.of.the.art results.

Foreground segmentation and shadow detection Segmentation has to deal with shadows to avoid distortions when detecting moving objects. Most approaches for shadow detection are typically restricted to penumbra shadows, *i.e.* shadows that exhibit low values of intensity and similar chromaticity values compared to the background, hence, such techniques cannot cope well with umbra shadows. These shadows exhibit different chromaticity than the background, and their intensity values are similar to those of any new object appearing in a scene. Consequently, umbra shadows are usually detected as part of moving objects. Chapter 7 contains the following contributions: combination of an invariant color cone model and an invariant gradient model to improve foreground segmentation and detection of potential shadows. Extending the shadow detection to cope with chromatic moving cast shadows, by grouping potential shadow regions and considering "a bluish effect", edge partitioning, temporal similarities between local gradient, and spatial similarities between chrominance angle and brightness distortions. Unlike other methods, the approach does not make any assumptions about camera location, surface geometries, surface textures, shapes and types of shadows, objects and background. Experimental results show promising performance and accuracy for surveillance scenarios with different shadowed materials and illumination conditions.

Shadow tracking Chapter 7 also propose tracking of both objects and shadows and establishing data association between them, and hereby achieve an enhancement of the chromatic shadow detection by recovering miss-detected shadows. To our best knowledge, this is the first attempt to introduce shadow tracking and apply this information to improve object and shadow detection. As a result, a more robust tracking is obtained by using mutual information and association of object and shadow, and an improvement of the segmentation for high level processes, such as detection, tracking and recognition, by avoiding shadows.

1.4 Datasets for Human Action Recognition

The development of new methods within video-based human activity recognition usually requires large amounts of video data and especially for the purpose of testing and validation there is a great need for large datasets with ground truth data. With public datasets of this kind it is also possible to directly compare new methods with state-of-the-art. Within action recognition especially two datasets have been widely used, namely the KTH dataset [42] and the Weizmann dataset [13]. This has allowed for direct comparisons between many methods but with recent publications reporting recognition rates of 100% for Weizmann and ∼96% for KTH, new and more challenging datasets are needed.

Within the last five years a large amount of datasets have been produced and made publicly available all targeting different aspects of human activity recognition. A trend in these datasets is more complex actions and activities sometimes involving multiple people and also multiple views of the scene. Another interesting feature with some of these new datasets is the release of implementations of baseline methods accompanying the datasets. This allows for a more thorough comparison and also significantly reduces the effort needed to compare already published methods to new datasets.

To give an overview the most popular datasets are listed in the following. This listing contains the central specifications like video resolution, number of cameras, number of subjects, actions performed, etc. A brief description of the content of the video and possible simplifications are also provided. There are other collections of datasets available but this presentation of datasets has a strong focus on action recognition and presents a precise and consistent listing of dataset characteristics. Furthermore, all these datasets have been used in the work of this thesis to evaluate and compare the developed approaches.

Weizmann The Weizmann dataset [42] contains 90 videos separated into 10 actions performed by 9 persons. The actions are: *bend*, *jumping-jacks*, *jump*, *jump-in-place*, *run*, *gallop-sideways*, *skip*, *walk*, *one-hand-waving* and *two-hands-waving*. The videos are captured of single actors with clean and simple backgrounds.

KTH The KTH dataset [13] consists of 6 different actions: *walking*, *jogging*, *running*, *boxing*, *clapping* and *waving*. These actions are performed in 4 different but well-controlled environments with clean and simple backgrounds by 25 different single actors, resulting in a total of 600 action instances.

CVC The CVC dataset [1] consists of 5 actors performing 7 actions: *walking*, *jogging*,

running (with horizontal and vertical two-way paths), *hand-waving*, *two-hands-waving*, *jump-in-place* and *bending*. The dataset is rated "semi-complex" and is interesting, since it has a textured background.

CMU The CMU dataset [21] is composed of 48 video sequences of five action classes: *jumping-jacks*, *pick-up*, *push-button*, *one-hand-waving* and *two-hands-waving*. The video data contains 110 actions which are down-scaled to 160×120 in resolution. This dataset has been recorded by a hand-held camera with moving people and vehicles in the background, and is known to be very challenging.

Hollywood 2 The Hollywood 2 dataset [27] is composed of video clips extracted from 69 Hollywood movies, and contains 12 classes of human actions and 10 classes of scenes distributed over 3669 video clips, resulting in approximately 20.1 hours of video in total. The 12 actions are: *AnswerPhone*, *DriveCar*, *Eat*, *FightPerson*, *GetOutCar*, *HandShake*, *HugPerson*, *Kiss*, *Run*, *SitDown*, *SitUp* and *StandUp*. In total, there are 1707 action samples divided into a training set (823 sequences) and a test set (884 sequences), where train and test sequences are obtained from different movies. The dataset contains approximately 150 samples per action class and 130 samples per scene class in training and test subsets, and intends to provide a comprehensive benchmark for human action recognition in realistic and challenging settings. Hollywood 2 is an expansion of the former Hollywood I human actions dataset.

UCF YouTube The YouTube dataset [25] is a collection of 1168 complex and challenging YouTube videos of 11 human actions categories: *basketball shooting*, *volleyball spiking*, *trampoline jumping*, *soccer juggling*, *horseback riding*, *cycling*, *diving*, *swinging*, *golf swinging*, *tennis swinging* and *walking (with a dog)*. The dataset has the following properties: a mix of steady cameras and shaky cameras, cluttered background, low resolution, and variation in object scale, viewpoint and illumination. The first four actions are easily confused with jumping, the next two may have similar camera motion, and all the swing actions share some common motions. Some actions are also performed with objects such as a horse, bike or dog. In order to remove the effect of similar backgrounds, the video sequences are organized into 25 relatively independent groups, where separate groups are either recorded in different environments or by different photographers.

MSR I The Microsoft research action dataset I (MSR I) [61] consists of 16 video sequences and a total of 63 actions: 14 *hand-clapping*, 24 *hand-waving* and 25 *boxing*, performed by 10 subjects. The sequences contain multiple types of action recorded in indoor and outdoor scenes with cluttered and moving backgrounds. Some sequences contain multiple actions performed by different people. Each video is of low resolution 320×240 with a frame rate of 15 frames per second, and their lengths are between 32 to 76 seconds.

Multi-KTH The Multi-KTH dataset [53] is a more challenging version of the KTH dataset. It contains 5 (except *running*) of the 6 KTH-actions, which have been recorded by a hand-held camera, with multiple simultaneous actors, a significant amount of camera motion, scale changes and a more realistic cluttered background.

IXMAS The INRIA Xmas Motion Acquisition Sequences (IXMAS) Multi-View Human Action Dataset [56]. It consists of 12 non-professional actors performing 13 daily-life actions 3 times: *check watch*, *cross arms*, *scratch head*, *sit down*, *get up*, *turn around*, *walk*, *wave*, *punch*, *kick*, *point*, *pick up* and *throw*. The dataset has been recorded by 5 calibrated and synchronized cameras, where the actors chose freely position and orientation, and consists of image sequences (390×291) and reconstructed 3D volumes ($64 \times 64 \times 64$ voxels), resulting in a total of 2340 action instances for all 5 cameras.

i3DPost the i3DPost Multi-View Human Action and Interaction Dataset [12]. This dataset, which has been generated within the Intelligent 3D Content Extraction and Manipulation for Film and Games EU funded research project, consists of 8 actors performing 10 different actions, where 6 are single actions: *walk*, *run*, *jump*, *bend*, *hand-wave* and *jump-in-place*, and 4 are combined actions: *sit-stand-up*, *run-fall*, *walk-sit* and *run-jump-walk*. Additionally, the dataset also contains 2 interactions: *handshake* and *pull*, and 6 basic facial expressions. The subjects have different body sizes, clothing and are of different sex and nationalities. The multi-view videos have been recorded by a 8 calibrated and synchronized camera setup in high definition resolution (1920×1080), resulting in a total of 640 videos (excluding videos of interactions and facial expressions). For each video frame a 3D mesh model of relatively high detail level ($20,000$-$40,000$ vertices and $40,000$-$80,000$ triangles) of the actor and the associated camera calibration parameters are available. The mesh models were reconstructed using a global optimization method proposed by Starck and Hilton [46].

Other datasets Among other less frequently used datasets are the Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion (HumanEva) [44], the CMU Motion of Body (MoBo) Database [14], the Multi-camera Human Action Video Dataset (MuHAVi) [2] and the KU Gesture Dataset [17]. For more information regarding available datasets please refer to [30].

1.5 Publications of the thesis

The publications that result from the work of this Ph.D. thesis are listed below. The publications listed in bold appears directly in this thesis as chapters 2 to 7.

Journal papers

1. **I. Huerta, M.B. Holte, T.B. Moeslund and J. Gonzàlez. Shadow Tracking for Improved Detection and Removal of Chromatic Moving Shadows. Submitted to *Transactions on Image Processing, IEEE Signal Processing Society*, 2011.**

2. **M.B. Holte, B. Chakraborty, J. Gonzàlez and T.B. Moeslund. A Local 3D Motion Descriptor for Multi-View Human Action Recognition from 4D Spatio-Temporal Interest Points. Submitted to *Journal of Selected***

***Topics in Signal Processing, IEEE Signal Processing Society*, 2011 (Decision: major revision).**

3. **M.B. Holte, C. Tran, M.M. Trivedi and T.B. Moeslund. Human 3D Body Modeling, Pose Estimation and Activity Recognition from Multi-View Videos: Comparative Explorations of Recent Developments. Submitted to *Journal of Selected Topics in Signal Processing, IEEE Signal Processing Society*, 2011 (Decision: minor revision).**

4. **B. Chakraborty, M.B. Holte, T.B. Moeslund and J. Gonzàlez. Selective Spatio-Temporal Interest Points. In *Computer Vision and Image Understanding, Elsevier*, doi:10.1016/j.cviu.2011.09.010, November 2011.**

5. **M.B. Holte, T.B. Moeslund and P. Fihl. View-Invariant Gesture Recognition using 3D Optical Flow and Harmonic Motion Context. In *Computer Vision and Image Understanding, Elsevier*, vol. 114, no. 11, pages 1353–1361, December 2010.**

6. M.B. Holte, T.B. Moeslund and P. Fihl. View invariant gesture recognition using the CSEM SwissRanger SR-2 camera. In *International Journal of Intelligent Systems Technologies and Applications, Inderscience Publishers*, vol. 5, no. 3/4, pages 295–303, November 2008.

Papers in lecture notes

7. **P. Fihl, M.B. Holte and T.B. Moeslund. Motion Primitives and Probabilistic Edit Distance for Action Recognition. In *Gesture-Based Human-Computer Interaction and Simulation, Lecture Notes in Computer Science, vol. 5085, Springer Berlin/Heidelberg*, January 2009.**

Peer reviewed conference papers

8. M.B. Holte, C. Tran, M.M. Trivedi and T.B. Moeslund. Human Action Recognition using Multiple Views: A Comparative Perspective on Recent Developments. In *ACM Multimedia Joint Workshop on Human Gesture and Behavior Understanding, Association for Computing Machinery, Scottsdale, Arizona, USA*, December 2011.

9. B. Chakraborty, M.B. Holte, T.B. Moeslund and J. Gonzàlez. A Selective Spatio-Temporal Interest Point Detector for Human Action Recognition in Complex Scenes. In *IEEE International Conference on Computer Vision, Barcelona, Spain*, November 2011.

10. M.B. Holte, T.B. Moeslund, N. Nikolaidis and I. Pitas. 3D Human Action Recognition for Multi-View Camera Systems. In *IEEE Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, Hangzhou, China*, May 2011.

11. I. Huerta, M.B. Holte, T.B. Moeslund and J. Gonzàlez. Detection and Removal of Chromatic Moving Shadows in Surveillance Scenarios. In *IEEE International Conference on Computer Vision, Kyoto, Japan*, September 2009 (4^{th} most viewed and top ranked paper from ICCV 2009 on www.sciweaver.org).

12. M.B. Holte, T.B. Moeslund and P. Fihl. Fusion of Range and Intensity Information for View Invariant Gesture Recognition. In *IEEE Computer Vision and Pattern Recognition Workshop on Time of Flight-based Computer Vision, Anchorage, AK, USA*, June 2008.

13. M.B. Holte and T.B. Moeslund. View Invariant Gesture Recognition using 3D Motion Primitives. In *IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, NV, USA*, April 2008.

14. M.B. Holte, T.B. Moeslund and P. Fihl. View Invariant Gesture Recognition using the CSEM SwissRanger SR-2 Camera. In *Dynamic 3D Imaging workshop, Heidelberg, Germany*, September 2007.

15. P. Fihl, M.B. Holte and T.B. Moeslund. Motion Primitives for Action Recognition. In *International Workshop on Gesture in Human-Computer Interaction and Simulation, Lisbon, Portugal*, May 2007.

16. P. Fihl, M.B. Holte, T.B. Moeslund and L. Reng. Action Recognition Using Motion Primitives and Probabilistic Edit Distance. In *IEEE Articulated Motion and Deformable Objects, Andratx, Mallorca, Spain, Springer-Verlag Berlin/Heidelberg*, July 2006.

Non-reviewed conference papers

17. T.B. Moeslund, P. Fihl and M.B. Holte. Action Recognition using Motion Primitives. In *Danish Conference on Pattern Recognition and Image Analysis*, Copenhagen, August 2006.

Technical reports

18. M.B. Holte and T.B. Moeslund. Introduction to the CSEM SwissRanger Camera. In *Technical Report CVMT-07-04, Laboratory of Computer Vision and Media Technology, Aalborg University, Denmark*, 2007.

19. M.B. Holte and T.B. Moeslund. Gesture Recognition using a Range Camera. In *Technical Report CVMT-07-01, Laboratory of Computer Vision and Media Technology, Aalborg University, Denmark*, 2007.

20. P. Fihl, M.B. Holte, T.B. Moeslund and L. Reng. Action Recognition in Semi-synthetic Images using Motion Primitives. In *Technical Report CVMT-06-01, Laboratory of Computer Vision and Media Technology, Aalborg University, Denmark*, 2006.

References

[1] The CVC dataset is available at http://iselab.cvc.uab.es/files/Tools/CvcAction-DataSet/index.htm.

[2] MuHAVi dataset instructions at http://dipersec.king.ac.uk/MuHAVi-MAS.

[3] J.K. Aggarwal and S. Park. Human Motion: Modeling and Recognition of Actions and Interactions. In *3DPVT*, 2004.

[4] A.A. Alatan, Y. Yemez, U. Güdükbay, X. Zabulis, K. Müller, C.E. Erdem, C. Weigel, and A. Smolic. Scene representation technologies for 3dtv - a survey. *EEE Trans. Circuits Syst. Video Techn.*, 17(11):1587–1605, 2007.

[5] A.A. Alonso, R.D Rosa, L.D. Val, M.I. Jimenez, and S. Franco. A robot controlled by blinking for ambient assisted living. *Proceedings of the 10th International Work-Conference on Artificial Neural Networks: Part II: Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, 2009.

[6] J. Assfalg, M. Bertini, C. Colombo, A.D. Bimbo, and W. Nunziati. Semantic annotation of soccer videos: automatic highlights identification. *CVIU*, 92(2-3), 2003.

[7] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, 2002.

[8] Y. Chen, L.B. Smith, S. Hongwei, A.F. Pereira, and T. Smith. Active information selection: Visual attention through the hands. *IEEE Transactions on Autonomous Mental Developmen*, 1(2):141–151, 2009.

[9] S.Y. Cheng and M.M. Trivedi. Turn-intent analysis using body pose for intelligent driver assistance. *IEEE Pervasive Computing*, 5(4):28–37, 2006.

[10] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.

[11] J. Geigel and M. Schweppe. Motion capture for realtime control of virtual actors in live, distributed, theatrical performances. In *FG*, 2011.

[12] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, and I. Pitas. The i3dpost multi-view and 3d human action/interaction database. In *CVMP*, 2009. The i3DPost dataset is available at http://kahlan.eps.surrey.ac.uk/i3dpost_action/data.

[13] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *PAMI*, 29(12):2247–2253, 2007. The Weizmann dataset is available at http://www.wisdom.weizmann.ac.il/∼vision/ SpaceTime Actions.html.

[14] R. Gross and J. Shi. The cmu motion of body (mobo) database. In *Techical Report*, 2001.

[15] M.B. Holte, T.B. Moeslund, N. Nikolaidis, and I. Pitas. 3d human action recognition for multi-view camera systems. In *3DIMPVT*, 2011.

[16] I. Huerta, M. Holte, T.B. Moeslund, and J. González. Detection and removal of chromatic moving shadows in surveillance scenarios. In *ICCV*, 2009.

[17] B.-W. Hwang, S. Kim, and S.-W. Lee. A fullbody gesture database for automatic gesture recognition. In *FG*, 2006. The KU Gesture Dataset is available at http://gesturedb.korea.ac.kr.

[18] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007.

[19] X. Ji and H. Liu. Advances in view-invariant human motion analysis: A review. *Trans. Sys. Man Cyber Part C*, 40(1):13–24, 2010.

[20] X. Ji, H. Liu, Y. Li, and D. Brown. Visual-Based View-Invariant Human Motion Analysis: A Review. In *Knowledge-Based Intelligent Information and Engineering Systems*, volume 5177 of *LNCS*. Springer Berlin/Heidelberg, 2008.

[21] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *ICCV*, 2007. The CMU dataset instructions are available at http://www.cs.cmu.edu/ ∼yke/video/#Dataset.

[22] J. Kilner, J.-Y. Guillemaut, and A. Hilton. 3d action matching with key-pose detection. In *ICCV Workshops*, 2009.

[23] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003.

[24] H. Lee and J. H. Kim. An hmm-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10), 1999.

[25] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *CVPR*, 2009. The YouTube dataset is available at http://www.cs.ucf.edu/ ∼liujg/YouTube_Action_dataset.html.

[26] J. Liu and M. Shah. Learning human actions via information maximization. In *CVPR*, 2008.

[27] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009. The Hollywood 2 dataset is available at http://www.irisa.fr/vista/ actions/hollywood2.

[28] T.B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *CVIU*, 81(3):231?268, 2001.

[29] T.B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 104(2-3):90–126, 2006.

[30] T.B. Moeslund, A. Hilton, V. Krüger, and L. Sigal. Visual analysis of humans: Looking at people. In *Springer Verlag, Berlin/Heidelberg*, 2011.

[31] L. Muendermann, S. Corazza, A.M. Chaudhari, T.P. Andriacchi, A. Sundaresan, and R. Chellappa. Measuring human movement for biomechanical applications using markerless motion capture. In *Proceeding of SPIE Three-Dimensional Image Capture and Applications*, 2006.

[32] E. Murphy-Chutorian and M.M. Trivedi. 3d tracking and dynamic analysis of human head movements and attentional targets. In *IEEE/ACM Int'l. Conf. on Distributed Smart Cameras*, 2008.

[33] E. Murphy-Chutorian and M.M. Trivedi. Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness. *IEEE Transactions on Intelligent Transportation Systems*, 2010.

[34] A. Oikonomopoulos, I. Patras, and M. Pantic. Spatiotemporal salient points for visual recognition of human actions. *SMC-B*, 36(3):710–719, 2006.

[35] S. Park and M.M. Trivedi. Understanding Human Interactions with Track and Body Synergies (TBS) Captured from Multiple Views. *Computer Vision and Image Understanding*, 111(1):2–20, 2008.

[36] S. Pehlivan and P. Duygulu. A new pose-based representation for recognizing actions from multiple cameras. *CVIU*, 115:140–151, 2011.

[37] R. Poppe. Vision-based human motion analysis: An overview. *CVIU*, 108(1-2):4–18, 2007.

[38] R. Poppe. A survey on vision-based human action recognition. *IVC*, 28(6):976–990, 2010.

[39] A.B. Postawa, M. Kleinsorge, J. Krueger, and G. Seliger. Automated image based recognition of manual work steps in the remanufacturing of alternators. *Advances in Sustainable Manufacturing*, 5:209–214, 2011.

[40] J. Radmer and J. Krueger. Depth data-based capture of human movement for biomechanical application in clinical rehabilitation use. In *5th International Symposium on Health Informatics and Bioinformatics*, 2010.

[41] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. Fall detection from human shape and motion history using video surveillance. In *21st International Conference on Advanced Information Networking and Applications Workshops*, 2007.

[42] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, 2004. The KTH dataset is available at http://www.nada.kth.se/cvap/actions.

[43] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011.

[44] L. Sigal and M.J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. In *Techniacl Report*, 2006. The HumanEva dataset is available at http://vision.cs.brown.edu/humaneva.

[45] R. Souvenir and J. Babbs. Learning the viewpoint manifold for action recognition. In *CVPR*, 2008.

[46] J. Starck and A. Hilton. Surface capture for performance based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31, 2007.

[47] C. Tran, A. Doshi, and M.M. Trivedi. Pedal errors prediction by driver foot gesture analysis: A vision-based inquiry. In *IEEE Intelligent Vehicle Symposium*, 2011.

[48] C. Tran and M.M. Trivedi. Driver assistance for 'keeping hands on the wheel and eyes on the road. In *IEEE International Conference on Vehicular Electronics and Safety*, 2009.

[49] C. Tran and M.M. Trivedi. Introducing XMOB: Extremity Movement Observation Framework for Upper Body Pose Tracking in 3D. In *IEEE International Symposium on Multimedia*, 2009.

[50] M.M. Trivedi and S.Y. Cheng. Holistic sensing and active displays for intelligent driver support systems. In *IEEE Computer Magazine*, 2007.

[51] M.M. Trivedi, S.Y. Cheng, E. Childers, and S. Krotosky. Occupant posture analysis with stereo and thermal infrared video: Algorithms and experimental evaluation. *IEEE Transactions on Vehicular Technology, Special Issue on In-Vehicle Vision Systems*, 53(6), 2004.

[52] M.M. Trivedi, K.S. Huang, and I. Mikic. Dynamic context capture and distributed video arrays for intelligent spaces. *IEEE Trans. on Systems, Man and Cybernetics, Part A*, 35(1):145–163, 2005.

[53] H. Uemura, S. Ishikawa, and K. Mikolajczyk. Feature tracking and motion compensation for action recognition. In *BMVC*, 2008. The Multi-KTH dataset is available at http://www.openvisor.org/ video_details.asp?idvideo=303.

[54] A. Utasi and C. Benedek. A 3-d marked point process model for multi-view people detection. In *CVPR*, 2011.

[55] A. Waibel, R. Stiefelhagen, R. Carlson, J. Casas, J. Kleindienst, L. Lamel, O. Lanz, D. Mostefa, M. Omologo, F. Pianesi, L. Polymenakos, G. Potamianos, J. Soldatos, G. Sutschet, and J. Terken. Computers in the human interaction loop. In *Handbook of Ambient Intelligence and Smart Environments, Springer*, 2010.

[56] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *CVIU*, 104(2):249–257, 2006. The IXMAS dataset is available at http://4drepository.inrialpes.fr/public/viewgroup/6.

[57] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *INRIA Report*, RR-7212:54–111, 2010.

[58] N. Werghi. Segmentation and modeling of full human body shape from 3-d scan data: A survey. *TSMC-C*, 37(6):1122–1136, 2007.

[59] G. Willems, T. Tuytelaars, and L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, 2008.

[60] S.F. Wong and R. Cipolla. Extracting spatiotemporal interest points using global information. In *ICCV*, 2007.

[61] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. In *CVPR*, 2009. The MSR dataset is available at http://research.microsoft.com/en-us/um/people/zliu/ActionRecoRsrc.

Chapter 2

2D Human Gesture Recognition

This chapter consists of the paper "Motion Primitives and Probabilistic Edit Distance for Action Recognition" [A]. The paper presents a key-frame based method to recognize arm gestures. The gestures represent typical signaling between people over long distances. The method uses traning data which is synthezised with a computer graphics model of a human and achieve good recognition results on real test video. References [B-E] descibe intermediate work resulting in the final outcome in [A].

References

A. P. Fihl, M.B. Holte and T.B. Moeslund. Motion Primitives and Probabilistic Edit Distance for Action Recognition. In *Gesture-Based Human-Computer Interaction and Simulation, Lecture Notes in Computer Science, vol. 5085, Springer Berlin/Heidelberg*, January 2009.

B. P. Fihl, M.B. Holte and T.B. Moeslund. Motion Primitives for Action Recognition. In *International Workshop on Gesture in Human-Computer Interaction and Simulation, Lisbon, Portugal*, May 2007.

C. P. Fihl, M.B. Holte, T.B. Moeslund and L. Reng. Action Recognition Using Motion Primitives and Probabilistic Edit Distance. In *IEEE Articulated Motion and Deformable Objects, Andratx, Mallorca, Spain, Springer-Verlag Berlin/Heidelberg*, July 2006.

D. T.B. Moeslund, P. Fihl and M.B. Holte. Action Recognition using Motion Primitives. In *Danish Conference on Pattern Recognition and Image Analysis*, Copenhagen, August 2006.

E. P. Fihl, M.B. Holte, T.B. Moeslund and L. Reng. Action Recognition in Semi-synthetic Images using Motion Primitives. In *Technical Report CVMT-06-01, Laboratory of Computer Vision and Media Technology, Aalborg University, Denmark*, 2006.

Motion Primitives and Probabilistic Edit Distance for Action Recognition

P. Fihl, M.B. Holte and T.B. Moeslund

Abstract

The number of potential applications has made automatic recognition of human actions a very active research area. Different approaches have been followed based on trajectories through some state space. In this paper we also model an action as a trajectory through a state space, but we represent the actions as a sequence of temporal isolated instances, denoted primitives. These primitives are each defined by four features extracted from motion images. The primitives are recognized in each frame based on a trained classifier resulting in a sequence of primitives. From this sequence we recognize different temporal actions using a probabilistic Edit Distance method. The method is tested on different actions with and without noise and the results show recognition rates of 88.7% and 85.5%, respectively.

2.1 Introduction

Automatic recognition of human actions is a very active research area due to its numerous applications. As opposed to earlier the current trend is not as much on first reconstructing the human and the pose of his/her limbs and *then* do the recognition on the joint angle data, but rather to do the recognition directly on the image data, e.g., silhouette data [20, 21, 23] or spatio-temporal features [1, 4, 15].

Common for these approaches is that they represent an action by image data from all frames constituting the action, e.g., by a trajectory through some state-space or a spatio-temporal volume. This means that the methods in general require that the applied image information can be extracted reliably in every single frame. In some situations this will not be possible and therefore a different type of approach has been suggested. Here an action is divided into a number of smaller temporal sequences, for example movemes [6], atomic movements [7], states [5], dynamic instants [16], examplars [11], behaviour units [9], and key-frames [8]. The general idea is that approaches based on finding smaller units will be less sensitive compared to approaches based on an entire sequence of information.

For some approaches the union of the units represents the entire temporal sequence, whereas for other approaches the units represent only a subset of the original sequence. In Rao *et al.* [16] dynamic hand gestures are recognized by searching a trajectory in 3D space (x and y-position of the hand, and time) for certain dynamic instants. Gonzalez *et al.* [8] look for key-frames for recognizing actions, like walking and running. Approaches where the entire trajectory (one action) is represented by a number of subsequences are Barbic *et al.* [2] for full body motion, where probabilistic PCA is used for finding transitions between different behaviors, and Bettinger *et al.* [3] where likelihoods are used to separate a trajectory into sub-trajectories. These sub-trajectories are modeled by Gaussian distributions each corresponding to a temporal primitive.

2.2 Paper content and system design

In this paper we address action recognition using temporal instances (denoted primitives) that only represent a subset of the original sequence. That is, our aim is to recognize an action by recognizing only a few primitives as opposed to recognition based on the entire sequence (possibly divided into sub-trajectories).

Our approach is based on the fact that an action will always be associated with a movement, which will manifest itself as temporal changes in the image. So by measuring the temporal changes in the image the action can be inferred. We define primitives as temporal instances with a significant change and an action is defined as a set of primitives. This approach allows for handling partly corrupted input sequences and, as we shall see, does not require the lengths, the start point, nor the end point to be known, which is the case in many other systems.

Measuring the temporal changes can be done in a number of ways. We aim at primitives that are as independent on the environment as possible. Therefore, we do not rely on figure-ground segmentation using methods like background subtraction or personalized models etc. Instead we define our primitives based on image subtraction. Image subtrac-

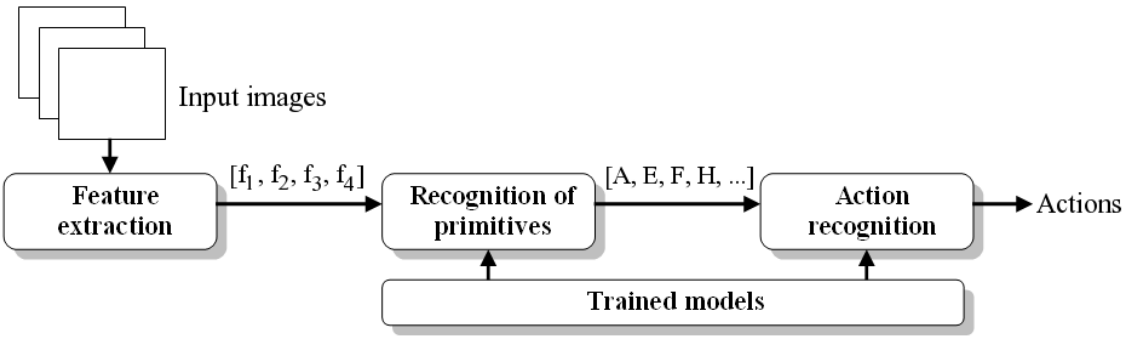


Figure 2.1: System overview.

tion has the benefit that it measures the change in the image over time and can handle very large changes in the environment.

Concretely we represent our primitives by four features extracted from a motion-image (found by image subtraction). In each frame the primitive, if any, that best explains the observed data is identified. This leads to a discrete recognition problem since a video sequence will be converted into a string containing a sequence of symbols, each representing a primitive. After pruning the string a probabilistic Edit Distance classifier is applied to identify which action best describes the pruned string. The system is illustrated in figure 2.1.

The actions that we focus on in this work are five one-arm gestures, but the approach can with some modifications be generalized to body actions. The actions are inspired by [10] and can be seen in figure 2.2.

The paper is structured as follows. In section 2.3 we describe how our features are extracted. In section 2.4 we describe how we recognize the primitives, and in section 2.5 we describe how we recognize the actions. In section 2.6 the approach is evaluated on a number of actions and in section 2.7 the approach is discussed.

2.3 Feature extraction

Even though image subtraction only provides crude information it has the benefit of being rather independent to illumination changes and clothing types and styles. Furthermore, no background model or person model is required. However, difference images suffer from "shadow effects" and we therefore apply double difference images, which are known to be more robust [22]. The idea is to use three successive images in order to create two difference images. These are thresholded and ANDed together. This ensures that only pixels that have changed in both difference images are included in the final output. The motion extraction process is illustrated in figure 2.3.

Multiple steps between the three successive images used to generate the double difference image have been investigated (frames 1-2-3, frames 1-3-5, and frames 1-4-7, etc.). The approach is rather invariant to this choice, i.e., invariant to the frame-rate and the execution speed of the actions. Frames 1-3-5 are used in this work.

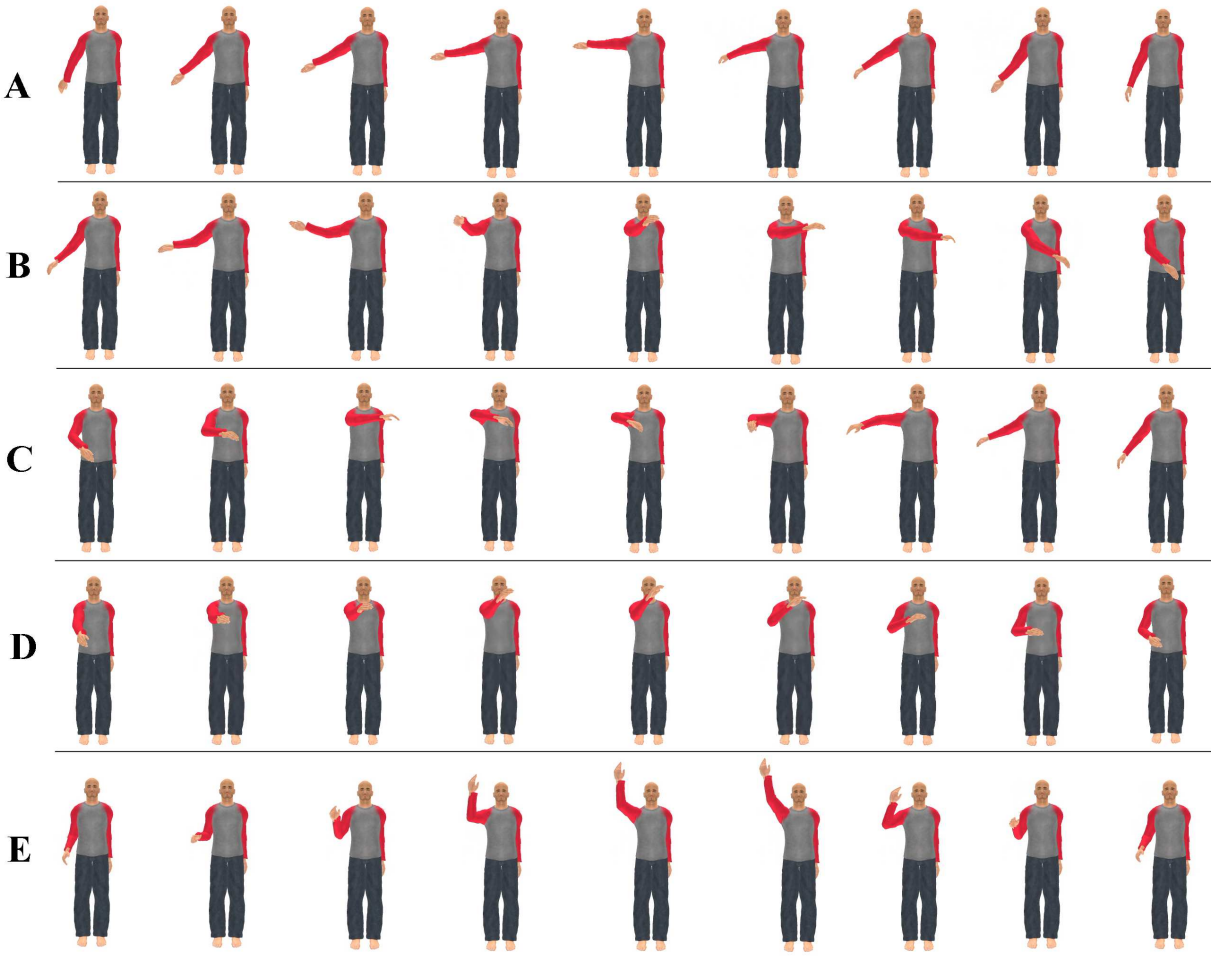


Figure 2.2: Samples from the five actions. The following describes the actions as seen from the person performing the action. **A - Point right:** A stretched arm is raised to a horizontal position pointing right, and then lowered down. **B - Move left:** A stretched arm is raised to a horizontal position pointing right. The arm is then moved in front of the body ending at the right shoulder, and then lowered down. **C - Move right:** Right hand is moved up in front of the left shoulder. The arm is then stretched while moved all the way to the right, and then lowered down. **D - Move closer:** A stretched arm is raised to a horizontal position pointing forward while the palm is pointing upwards. The hand is then drawn to the chest, and lowered down. **E - Raise arm:** The arm is moved along the side of the person, stretched above the head, and then lowered again.

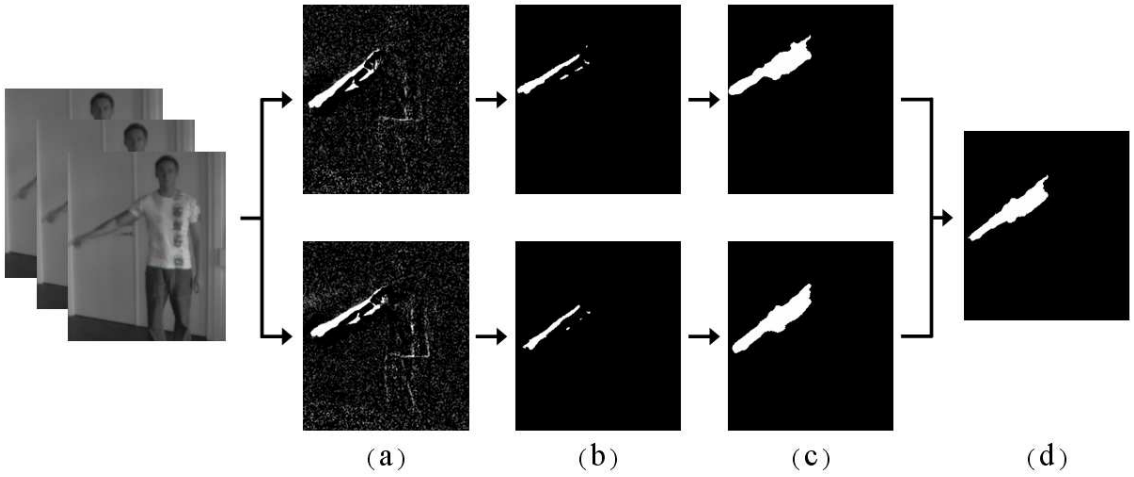


Figure 2.3: An illustration of the motion extraction process. (a) Difference images are calculated from a set of three input frames yielding noisy gray scale images. (b)+(c) The hysteresis thresholds T_1 and T_2 are applied. (d) The two thresholded images from (c) are ANDed together resulting in a single connected motion-cloud.

When doing arm gestures the double difference image will roughly speaking contain a "motion-cloud". However, noise will also be present. Either from other movements, e.g., the clothes on the upper body when lifting the arm (false positives), or the motion-cloud will be split into a number of separate blobs, e.g., due to the shirt having a uniform color (false negatives). Since the two noise sources "work against each other", it is difficult to binarize the difference image. We therefore apply a hysteresis principle consisting of two thresholds T_1 and T_2 with $T_1 > T_2$. For all difference pixels above T_1 we initiate a region growing procedure which continues to grow until the pixel values falls below T_2, see figure 2.4.

The resulting connected motion components are further sorted with respect to their size to obtain robustness towards noise. This hysteresis threshold helps to ensure that noisy motion-clouds are not broken up into multiple fragments and at the same time eliminates small noisy motion blobs. The result is one connected motion-cloud.

We model the motion-cloud compactly by an ellipse. The length and orientation of the axes of the ellipse are calculated from the Eigen-vectors and Eigen-values of the covariance

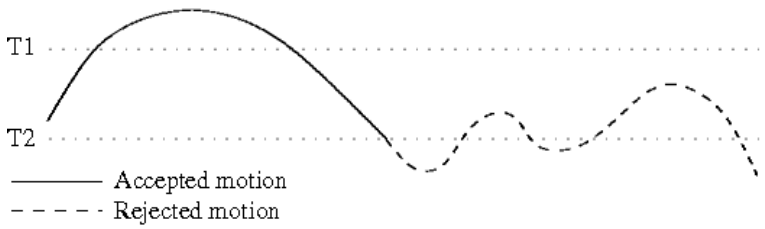


Figure 2.4: An illustration of the hysteresis with an upper threshold T_1 and a lower threshold T_2. The figure illustrates the advantage of the hysteresis, where most of the "motion-blob" of interest is accepted while the smaller "noise-blobs" are rejected.

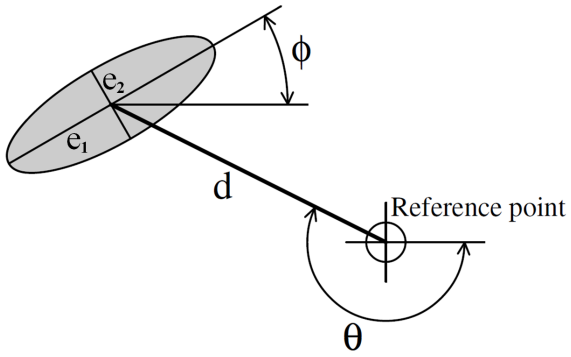


Figure 2.5: An illustration of the features used for describing the motion-cloud. See text for details.

matrix defined by the motion pixels. We use four features to represent the motion cloud. They are independent of image size and the person's size and position in the image. To ensure the scale invariance two of the features are defined with respect to a reference point currently defined manually as the center of gravity of the person. The features are illustrated in figure 2.5 and defined as follows. Feature 1 is the eccentricity of the ellipse defined as the ratio between the axes of the ellipse (e_2/e_1). Feature 2 is the orientation of the ellipse defined as the angle between the x-axis of the image and the major axis of the ellipse (ϕ). Feature 3 is the size of the ellipse defined as the ratio between the length of the major axis and the distance from the reference point to the center of the ellipse (e_1/d). Feature 4 is the angle between the x-axis of the image through the reference point and the line from the center of the ellipse to the reference point (θ).

2.4 Recognition of primitives

Each incoming frame is represented by the four extracted features described above. This feature vector is then classified as a particular primitive or as noise with a Mahalanobis classifier. From a set of training examples we extract representative feature vectors for each primitive. The primitives are then formed by the mean and covariance of the representative feature vectors, see below. The four features are not equally important and therefore weighted in accordance with their importance in classification. Experiments yielded features 2 and 4 as the most descriminative and feature 1 as the least descriminative. This gives the following classifier for recognizing a primitive at time t:

$$\text{Primitive}(t) = \arg\min_i \left[(\vec{W} \cdot (\vec{f_t} - \vec{p_i}))^T \Pi_i^{-1} (\vec{W} \cdot (\vec{f_t} - \vec{p_i})) \right] \tag{2.1}$$

where $\vec{f_t}$ is the feature vector estimated at time t, $\vec{p_i}$ is the mean vector of the ith primitive, Π_i is the covariance matrix of the ith primitive, and $\vec{W}$ contains the weights and are included as an element-wise multiplication.

The classification of a sequence can be viewed as a trajectory through the 4D feature space where, at each time-step, the closest primitive (in terms of Mahalanobis distance) is found. To reduce noise in this process we introduce a minimum Mahalanobis distance in order

for a primitive to be considered in the first place. Furthermore, to reduce the flickering observed when the trajectory passes through a border region between two primitives we introduce a hysteresis threshold. It favors the primitive recognized in the preceding frame over all other primitives by modifying the individual distances. The classifier hereby obtains a "sticky" effect, which handles a large part of the flickering.

After processing a sequence the output will be a string with the same length as the sequence. An example is illustrated in equation 2.2. Each letter corresponds to a recognized primitive (see figure 2.7) and Ø corresponds to time instances where no primitives are below the minimum required Mahalanobis distance. The string is pruned by first removing 'Ø's, isolated instances, and then all repeated letters, see equation 2.3. A weight is generated to reflect the number of repeated letters (this is used below).

$$\text{String} = \{Ø, Ø, B, B, B, B, B, E, A, A, F, F, F, F, Ø, D, D, G, G, G, G, Ø\} \quad (2.2)$$

$$\text{String} = \{B, A, F, D, G\} \quad (2.3)$$

$$\text{Weights} = \{5, 2, 4, 2, 4\} \quad (2.4)$$

2.4.1 Learning models for the primitives

In order to recognize the primitives we need to have a prototypical representation of each primitive, i.e., a mean and covariance in the 4D feature space. As can be seen in figure 2.2 the actions are all fronto-parallel. Ongoing work aims to generalize this work by allowing for multiple viewpoints. One problem with multiple viewpoints is how to train the system - it will require a very large number of test sequences. Therefore we have captured all training data using a magnetic tracking system with four sensors. The sensors are placed at the wrist, at the elbow, at the shoulder, and at the upper torso (for reference). The hardware used is the Polhemus FastTrac [18] which gives a maximum sampling rate of 25Hz when using four sensors. The data is converted into four Euler angles: three at the shoulder and one at the elbow in order to make the data invariant to body size. An action corresponds to a trajectory through a 4D space spanned by the Euler angles.

The data is input to a commercial computer graphics human model, Poser [19], which then animates all captured data. This allows us to generate training data for any view point and to generate additional training data by varying the Euler angles (based on the training data) and varying the clothing of the model. Figure 2.6 shows a person with magnetic trackers mounted on the arm, two different visualizations of the 3D tracker data from Poser, and an example of the test data. Based on this synthetic training data we build a classifier for each primitive.

2.4.2 Defining the primitives

Defining the number of primitives and their characteristics ("human movement") is quite a significant optimization problem. We are aiming at automating this process [17], but in this work the definition of primitives was done manually.

The primitives are defined based on an evaluation of video sequences showing three different people performing the five actions. The criteria for defining the primitives are 1) that they represent characteristic and representative 3D configurations, 2) that their projected

Figure 2.6: An illustration of the different types of data used in the system. From left to right: 1) 3D tracker data is acquired from magnetic trackers mounted on persons who perform the five actions. 2) The tracker data is animated in Poser from a fronto-parallel view. 3) The tracker data can be animated from any view point with different clothings and models. 4) After training the primitives on semi-sythetic data we recognize actions in real video.

2D configurations contain a certain amount of fronto-parallel motion, and 3) that the primitives are used in the description of as many actions as possible, i.e., fewer primitives are required. In this way we find 10 primitives that can represent the five actions. Each primitive is appearing in several actions resulting in five to eight primitives for each action.

To obtain the prototypical representation we randomly select 20 samples of each primitive and render the appropriate motion capture data to get a computer graphics representation of that sample. The double difference images of these samples are calculated and each of the motion-clouds are represented by the four features. The 20 samples then yields a mean vector and a 4x4 covariance matrix for each primitive. In figure 2.7 the 10 primitives and their representations are visualized together with the letter denoting the primitive. We can use the computer generated version of the training samples in stead of the original real video since the resulting double difference images are comparable and with this approach we achieve the possibility of generating new training data in a fast and flexible way without recording new training video.

2.5 Recognition of actions

The result of recognizing the primitives is a string of letters referring to the known primitives. During a training phase a string representation of each action to be recognized is learned. The task is now to compare each of the learned actions (strings) with the detected string. Since the learned strings and the detected strings (possibly including errors!) will in general not have the same length, the standard pattern recognition methods will not suffice. We therefore apply the Edit Distance method [12], which can handle matching of strings of different lengths.

The edit distance is a well known method for comparing words or text strings, e.g., for spell-checking and plagiarism detection. It operates by measuring the distance between two strings in terms of the number of operations needed in order to transform one into the other. There are three possible operations: *insert* a letter from the other string,

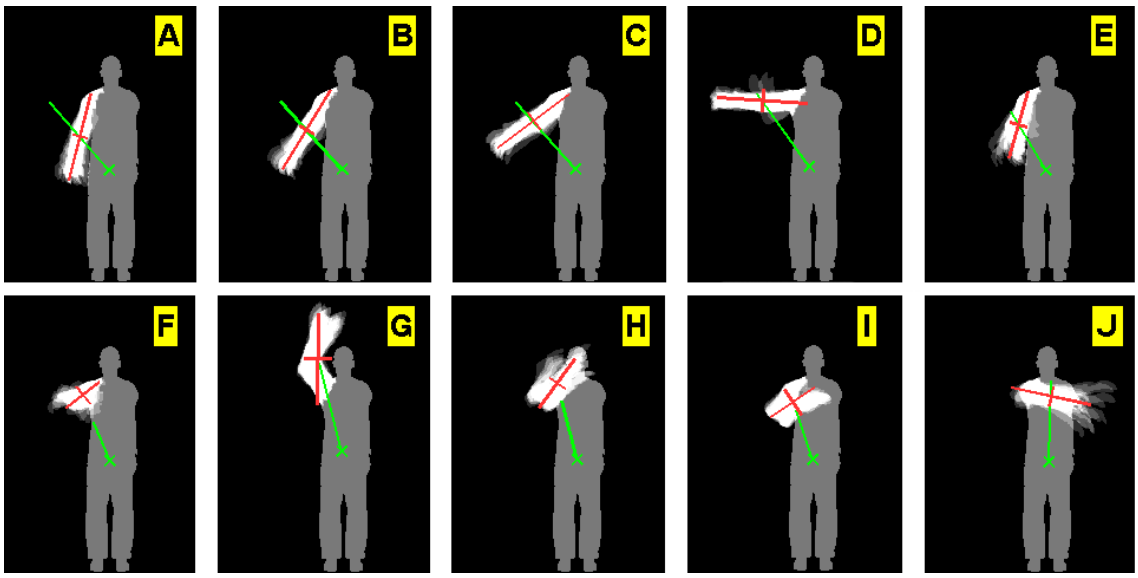


Figure 2.7: The figure of each primitive contains the silhouettes of the 20 samples added together which gives the gray silhouette. The 20 motion clouds from the double difference images of the samples are added on top of the silhouette as the white cloud. The figures furthermore illustrates the mean of the four features for each primitive by depicting the axes of the fitted ellipse and the distance and direction from the reference point to the motion cloud.

delete a letter, and *exchange* a letter by one from the other string. Whenever one of these operations is required in order to make the strings more similar, the score or distance is increased. The algorithm is illustrated in figure 2.8 where the strings *motions* and *octane* are compared.

The first step is initialization. The two strings are placed along the sides of the matrix, and increasing numbers are place along the borders beside the strings. Hereafter the matrix is filled cell by cell by traversing one column at a time. If the letters at row i and column j are the same then cell $c_{i,j}$ is assigned the same value as cell $c_{i-1,j-1}$. Otherwise cell $c_{i,j}$ is assigned the smallest value of the following three operations:

$$Insert: \quad c_{i-1,j} + cost \tag{2.5}$$

$$Delete: \quad c_{i,j-1} + cost \tag{2.6}$$

$$Exchange: \quad c_{i-1,j-1} + cost \tag{2.7}$$

In the original edit distance method the *cost* equals one.

Using these rules the matrix is filled and the value found at the bottom right corner is the edit distance required in order to map one string into the other, i.e., the distance between the two strings. The actual sequence of operations can be found by back-tracing the matrix. More than one path is often possible.

The edit distance is a deterministic method but by changing the cost of each of the three operations with respect to likelihoods it becomes a probabilistic method. The edit distance method has several variations that define cost functions in different ways, e.g. the Weighted Edit Distance where a cost function is defined for each operation or the

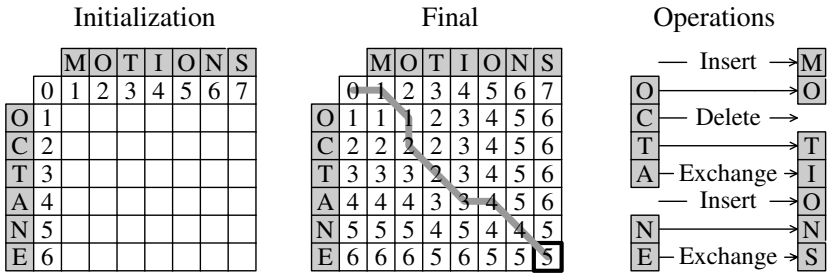


Figure 2.8: Measuring the distance between two strings using edit distance.

Needleman-Wunsch algorithm [14] where a cost matrix is used to define the similarity between the symbols (letters) of the applied set of symbols (alphabet). In stead of defining a fixed cost for an operation or each symbol-pair we define the cost of applying operations to a primitive based on the actual observations at any given time. The number of repetitions of a primitive to some extent represent the likelihood of that primitive being correct. This means, in terms of a cost function, that the cost of deleting or exchanging a primitive that have been observed multiple times should be increased with the number of observed repetitions. Concretely we incorporate the weights described above (see equation 2.4) into the edit distance method by increasing the cost of the *delete* and *exchange* operations by the weight multiplied by β (a scaling factor). The cost of *inserting* remains 1.

When the strings representing the actions are of different lengths, the method tends to favor the shorter strings. Say we have detected the string $\{B, C, D\}$ and want to classify it as being one of the two actions: $a_1 = \{J, C, G\}$ and $a_2 = \{A, B, C, D, H\}$. The edit distance from the detected string to the action-strings will be two in both cases. However, it seems more likely that the correct interpretation is that the detected string comes from a_2 in a situation where the start and end has been corrupted by noise. In fact, 2 out of 3 of the primitives have to be changed for a_1 whereas only 2 out of 5 have to be changed for a_2. We therefore normalize the edit distance by dividing the output by the length of the action-string, yielding 0.67 for a_1 and 0.2 for a_2, i.e., a_2 is recognized.

The above principle works for situations where the input sequence only contains one action (possibly corrupted by noise). In a real scenario, however, we will have sequences which are potentially much longer than an action and which might include more actions after each other. The action recognition problem is therefore formulated as for each action to find the substring in the detected string, which has the minimum edit distance. The recognized action will then be the action that has the substring with the overall minimum edit distance. Denoting the start point and length of the substring, s and l, respectively, we recognize the action present in the detected string as:

$$\text{Action} = \arg\min_{k,s,l} PED(\Lambda, k, s, l) \tag{2.8}$$

where k index the different actions, Λ is the detected string, and $PED(\cdot)$ is the probabilistic edit distance.

2.6 Results

2.6.1 Test setup

Two kinds of tests are conducted: one with known start and stop time of action execution, and another with "noise" added in the beginning and end of the sequences (unknown start time). By adding noise to the sequence we introduce the realistic problem of having no clear idea about when an action commences and terminates which would be the case in a real situation. To achieve a test scenario that resembles this situation we split the five actions into halves and add one of these half actions to the beginning and one to the end of each action to be processed by the system. The added half actions are chosen randomly resulting in unknown start and end point of the real action.

We use eleven test subjects, whom each performs each gesture 10 times. This leads to 550 sequences. The weighting of the features $\vec{W}$ are set to $\{1, 4, 2, 4\}$, and $\beta = 1/8$. $\vec{W}$ and β are determined through quantitative tests. A string representation of each action is found and since the shortest string contains five primitives and the longest eight primitives, we only perform the probabilistic edit distance calculation for substrings having the lengths $\in [3, 15]$.

2.6.2 Tests

	1.	2.	3.	4.	5.
1. Point right	**100**				
2. Move left	**6.4**	**90.9**		**2.7**	
3. Move right	**5.5**		**92.7**	**0.9**	**0.9**
4. Move closer		**2.7**	**1.8**	**70.9**	**23.6**
5. Raise arm				**10.9**	**89.1**

(a) Known start and stop time.

	1.	2.	3.	4.	5.
1. Point right	**99.1**		**0.9**		
2. Move left	**9.1**	**90.0**		**0.9**	
3. Move right	**7.3**		**90.0**	**2.7**	
4. Move closer	**0.9**	**4.5**	**1.8**	**62.7**	**30.0**
5. Raise arm	**1.8**	**1.8**		**10.9**	**85.5**

(b) Unknown start and stop time.

Figure 2.9: The confusion matrices for the recognition rates (in percent) without added noise (a) and with added noise (b). Zero values have been left out to ease the overview of the confusion.

The overall recognition rate for the test with known start time is 88.7%. In figure 2.9(a) the confusion matrix for the results is shown. As can be seen in the figure, most of the errors occur by miss-classification between the two actions: *move closer* and *raise arm*. The main reasons for this confusion are the similarity of the actions, the similarity of the primitives in these actions, and different performances of the actions of different test subjects (some do not raise their arm much when preforming the *raise arm* action). As can be seen in figure 2.2 both actions are performed along the side of the person when seen from the fronto-parallel view and differs mainly in how high the arm is raised. From figure 2.7 it can be seen that primitives 'F', 'G', 'H', and 'I' have similar angles between the reference point and the motion cloud and 'F', 'H' and 'I' also have similar orientation of the ellipse. These two features, which are the ones with highest weights, make these four primitives harder to distinguish.

Figure 2.9(b) shows the confusion matrix for the test results with noise. The overall recognition rate for this test is 85.5%. The errors are the same as before but with some few additional errors caused by the unknown start and end time of the actions.

2.7 Conclusion

In this paper we have presented an action recognition approach based on motion primitives as opposed to trajectories. Furthermore, we extract features from temporally local motion as opposed to background subtraction or another segmentation method relying on learned models and a relatively controlled environment. We hope this makes our approach less sensitive, but have still to prove so in a more comprehensive test.

The results are promising due to two facts. First, the models are generated from synthetic data (generated based on test subjects) while the test data are real data. In fact, the test data and training data are recorded several months apart, hence this is a real test of the generalization capabilities of the action recognition process. This means that we can expect to use the same scheme when learning models for the next incarnation of the system, which is aimed at view-invariant action recognition. Secondly, the system does not break down when exposed to realistic noise. This suggests that the approach taking has potential to be expanded into a real system setup, as opposed to a lab setup which is virtually always used when testing action recognition systems.

The primitives used in this work are found manually. This turned out to be quite an effort due to the massive amount of data and possibilities. Currently we are therefore working to automate this process [17]. Another ongoing activity is to avoid manually defining the reference point, see section 2.3, by using the face as a reference for the features [13].

References

[1] R.V. Babu and K.R. Ramakrishnan. Compressed domain human motion recognition using motion history information. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing"*, Hong Kong, April, 2003.

[2] J. Barbic, N.S. Pollard, J.K. Hodgins, C. Faloutsos, J-Y. Pan, and A. Safonova. Segmenting Motion Capture Data into Distinct Behaviors. In *Graphics Interface*, London, Ontario, Canada, May 17-19 2004.

[3] F. Bettinger and T.F. Cootes. A Model of Facial Behaviour. In *IEEE International Conference on Automatic Face and Gesture Recognition*, Seoul, Korea, May 17 - 19 2004.

[4] A. Bobick and J. Davis. The Recognition of Human Movement Using Temporal Templates. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.

[5] A.F. Bobick and J. Davis. A Statebased Approach to the Representation and Recognition of Gestures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(12):1325 – 1337, 1997.

[6] C. Bregler. Learning and Recognizing Human Dynamics in Video Sequences. In *Conference on Computer Vision and Pattern Recognition*, pages 568 – 574, San Juan, Puerto Rico, 1997.

[7] L. Campbell and A. Bobick. Recognition of Human Body Motion Using Phase Space Constraints. In *International Conference on Computer Vision*, Cambridge, Massachusetts, 1995.

[8] J. Gonzalez, J. Varona, F.X. Roca, and J.J. Villanueva. *aSpaces*: Action spaces for recognition and synthesis of human actions. In *AMDO*, pages 189–200, 2002.

[9] O.C. Jenkins and M.J. Mataric. Deriving Action and Behavior Primitives from Human Motion Data. In *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, pages 2551–2556, Lausanne, Switzerland, Sept., 2002.

[10] A. Just and S. Marcel. HMM and IOHMM for the Recognition of Mono- and Bi-Manual 3D Hand Gestures. In *ICPR workshop on Visual Observation of Deictic Gestures (POINTING04)*, Cambridge, UK, August 2004.

[11] A. Kale, N. Cuntoor, and R. Chellappa. A Framework for Activity-Specific Human Recognition. In *International Conference on Acoustics, Speech and Signal Processing*, Orlando, Florida, May 2002.

[12] V.I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848, 1965.

[13] T.B. Moeslund, J.S. Petersen, and L.D. Skalski. Face Detection Using Multiple Cues. In *Scandinavian Conference on Image Analysis*, Aalborg, Denmark, June 10-14 2007.

[14] S.B. Needleman and C.D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–53, 1970.

[15] Alonso Patron-Perez and I. Reid. A Probabilistic Framework for Recognizing Similar Actions using Spatio-Temporal Features. In *British Machine Vision Conference*, UK, Sep. 2007.

[16] C. Rao, A. Yilmaz, and M. Shah. View-Invariant Representation and Recognition of Actions. *Journal of Computer Vision*, 50(2):55 – 63, 2002.

[17] L. Reng, T.B. Moeslund, and E. Granum. Finding Motion Primitives in Human Body Gestures. In S. Gibet, N. Courty, and J.-F. Kamps, editors, *GW 2005*, number 3881 in LNAI, pages 133–144. Springer Berlin Heidelberg, 2006.

[18] *http://polhemus.com/*, January 2006.

[19] *http://www.poserworld.com/*, January 2006.

[20] D. Weinland, R. Ronfard, and E. Boyer. Free Viewpoint Action Recognition using Motion History Volumes. *Computer Vision and Image Understanding*, 104(2):249–257, 2006.

[21] A. Yilmaz and M. Shah. Actions Sketch: A Novel Action Representation. In *IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, June, 2005.

[22] K. Yoshinari and M. Michihito. A Human Motion Estimation Method using 3-Successive Video Frames. In *Int. Conf. on Virtual Systems and Multimedia*, Gifu, Japan, 1996.

[23] H. Yu, G.-M. Sun, W.-X. Song, and X. Li. Human Motion Recognition Based on Neural Networks. In *ICCCS*, Hong Kong, May 2005.

Chapter 3

3D Human Gesture Recognition

This chapter consists of the paper "View-Invariant Gesture Recognition using 3D Optical Flow and Harmonic Motion Context" [A]. The paper presents work that builds on the gesture recognition of chapter 2, and documents a direct and very interesting extension of the gesture recognition using motion primitives. The principle of motion primitives for gesture recognition is in this paper used to develop a view invariant method by use of a Time-of-flight range camera. References [B-G] descibe intermediate work resulting in the final outcome in [A].

References

A. M.B. Holte, T.B. Moeslund and P. Fihl. View-Invariant Gesture Recognition using 3D Optical Flow and Harmonic Motion Context. In *Computer Vision and Image Understanding, Elsevier*, vol. 114, no. 11, pages 1353–1361, December 2010.

B. M.B. Holte, T.B. Moeslund and P. Fihl. View invariant gesture recognition using the CSEM SwissRanger SR-2 camera. In *International Journal of Intelligent Systems Technologies and Applications, Inderscience Publishers*, vol. 5, no. 3/4, pages 295–303, November 2008.

C. M.B. Holte, T.B. Moeslund and P. Fihl. Fusion of Range and Intensity Information for View Invariant Gesture Recognition. In *IEEE Computer Vision and Pattern Recognition Workshop on Time of Flight-based Computer Vision, Anchorage, AK, USA*, June 2008.

D. M.B. Holte and T.B. Moeslund. View Invariant Gesture Recognition using 3D Motion Primitives. In *IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, NV, USA*, April 2008.

E. M.B. Holte, T.B. Moeslund and P. Fihl. View Invariant Gesture Recognition using the CSEM SwissRanger SR-2 Camera. In *Dynamic 3D Imaging workshop, Heidelberg, Germany*, September 2007.

F. M.B. Holte and T.B. Moeslund. Introduction to the CSEM SwissRanger Camera. In *Technical Report CVMT-07-04, Laboratory of Computer Vision and Media Technology, Aalborg University, Denmark*, 2007.

G. M.B. Holte and T.B. Moeslund. Gesture Recognition using a Range Camera. In *Technical Report CVMT-07-01, Laboratory of Computer Vision and Media Technology, Aalborg University, Denmark*, 2007.

View-Invariant Gesture Recognition using 3D Optical Flow and Harmonic Motion Context

M.B. Holte, T.B. Moeslund and P. Fihl

Abstract

This paper presents an approach for view-invariant gesture recognition. The approach is based on 3D data captured by a SwissRanger SR4000 camera. This camera produces both a depth map as well as an intensity image of a scene. Since the two information types are aligned, we can use the intensity image to define a region of interest for the relevant 3D data. This data fusion improves the quality of the motion detection and hence results in better recognition. The gesture recognition is based on finding motion primitives (temporal instances) in the 3D data. Motion is detected by a 3D version of optical flow and results in velocity annotated point clouds. The 3D motion primitives are represented efficiently by introducing motion context. The motion context is transformed into a view-invariant representation using spherical harmonic basis functions, yielding a harmonic motion context representation. A probabilistic Edit Distance classifier is applied to identify which gesture best describes a string of primitives. The approach is trained on data from one viewpoint and tested on data from a very different viewpoint. The recognition rate is 94.4% which is similar to the recognition rate when training and testing on gestures from the same viewpoint, hence the approach is indeed view-invariant.

3.1 Introduction

Automatic analysis of humans and their actions has received increasingly more attention in the last decade [20]. One of the areas of interest is recognition of human gestures for use in for example Human Computer Interaction.

Many different approaches to gesture recognition have been reported [19]. They apply a number of different segmentation, feature extraction, and recognition strategies. E.g. [25] and [28] extract and represent human gestures/actions by velocity histories of tracked keypoints and ballistic dynamics, respectively, while gestures are recognized, e.g., through Hidden Marcov Models (HMMs) [2, 26, 27] or Dynamic Baysian Networks (DBNs) [3, 35]. These methods are virtually all based on analyzing 2D data, i.e., images. A consequence of this is that approaches only analyze 2D gestures carried out in the image plane, which is only a projection of the actual gesture. As a result, the projection of the gesture will be dependent on the viewpoint, and not contain full information about the performed gesture. To overcome this shortcoming the use of 3D data has been introduced through the use of two or more cameras, see for example [5, 7]. In this way, e.g., the surface structure or a 3D volume of the person can be reconstructed, and thereby a more descriptive representation for gesture recognition can be established. We follow this line of work and also apply 3D data. To avoid the difficulties inherent to classical stereo approaches (the correspondence problem, careful camera placement and calibration) we instead apply a *Time-of-Flight* (ToF) range camera – SwissRanger SR4000. Each pixel in this camera directly provides a depth value (distance to object). Even though the technology in range cameras is still in its early days, e.g., resulting in low resolution data, the great potential of such sensors has already resulted in them being applied in a number of typical computer vision applications like face detection [6], face tracking [17], shape analysis [13, 21], robot navigation [24] and gesture-based scene navigation [29]. In [1] a survey of recent developments in ToF-technology are presented. It discusses applications of this technology for vision, graphics, and HCI.

The development of range cameras has progressed rapidly over the last few years, leading to the release of new and improved camera models from some of the main manufacturers: MESA Imaging [18], PMD Technologies [23] and 3DV Systems [33]. Recently, MESA Imaging released the new SwissRanger SR4000 range camera with higher frame rate (up to 54 fps) and resolution (176×144 pixels). 3DV Systems is aiming at a consumer class range camera with similar size and look as a regular web-camera and a integrated sensor capable of producing 1 mega pixels color images, while PMD Technologies made a camera version with improved operating range (up to 40 m) for e.g. pedestrian detection in cars.

The SwissRanger camera that we apply also provides an amplitude value corresponding to an intensity value for each pixel. This means that at any given time instant both a depth image and an intensity image are present. For some applications these two information types compliment each other and are therefore both used. For example in [21] where the objective is to segment planar surfaces in 3D (range) data, the edges in the intensity image are applied to improve the result. Similar benefits of applying both data types can be seen in [6, 17, 8]. We also apply both data types and will show how they compliment each other.

Applying 3D data allows for analysis of 3D gestures. However, we are still faced with the

problem that a user has to be fronto-parallel with respect to the camera. A few works have been reported without the assumption on the user being fronto-parallel. E.g. in [30] where 5 calibrated and synchronized cameras are used to acquire data (the publicly available IXMAS data set), which is further projected to 64 evenly spaced virtual cameras used for training. Actions are described in a view-invariant manner by computing $\mathcal{R}$ transform surfaces and manifold learning. Similarly, [7] use the same data set to compute motion history volumes, which are used to derive view-invariant motion descriptors in Fourier space. Another example is seen in [5] where 3D Human Body Shapes are used for view-independent identification of human body postures, which are trained and tested on another multi-camera dataset.

The need for multiple calibrated and synchronized cameras followed up by an exhaustive training phase for multiple viewpoints is obviously not desirable. Instead we aim at a view-invariant approach which is trained by examples from one camera viewpoint and able to recognize gestures from a very different viewpoint, say $\pm 45°$. Another issue we want to combat is the often used assumption of known start- and end points. That is, often the test data consists of N sequences where each sequence contains one and only one gesture. This obviously makes the problem easier and it favors a trajectory-based approach, where each gesture is represented as a trajectory though some state-space with known start and end point. For real-life scenarios the start and end point is normally not known. To deal with this issue we follow the notion of recognition through a set of primitives [10, 11, 34, 37]. Concretely, we define a primitive as a time instance with significant 3D motion.

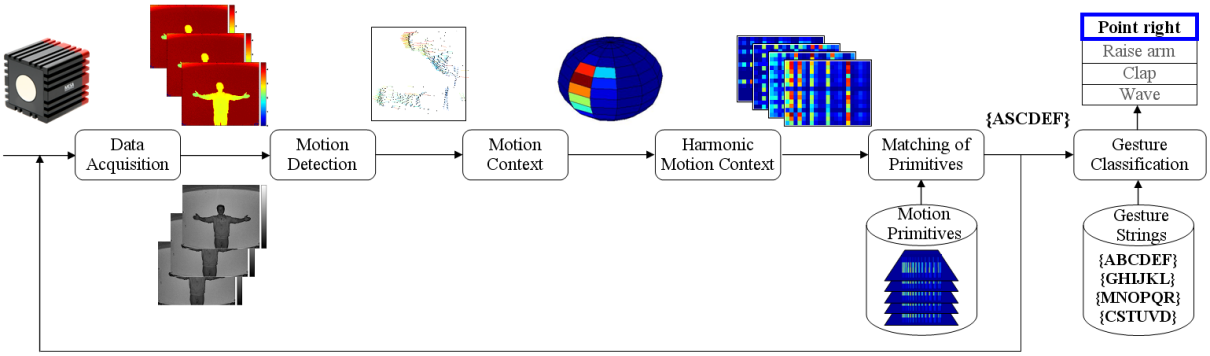


Figure 3.1: An overview of the range and intensity based gesture recognition system. Note that the feedback loop illustrates that a number of frames are processed before recognition of gestures commences.

3.1.1 Our approach

So, we represent gestures as an ordered sequence of *3D motion primitives* (temporal instances). We focus on arm gestures and therefore only segment the arms (when they move) and hereby suppress the rest of the (irrelevant) body information. Concretely we extract the moving arms using a 3D version of *optical flow* to produce *velocity annotated point clouds* and represent this data efficiently by their *motion context*. The motion context is an extended version of the regular shape context [4], and represents the velocity

annotated point cloud by using both the location of motion, together with the amount of motion and its direction. We make the primitives invariant to rotation around the vertical axis by re-representing the motion context using *spherical harmonic basis functions*, yielding a *harmonic motion context* representation. In each frame the primitive, if any, which best explains the observed data is identified. This leads to a discrete recognition problem since a video sequence of range data will be converted into a string containing a sequence of symbols, each representing a primitive. After pruning the string a *probabilistic Edit Distance classifier* is applied to identify which gesture best describes the pruned string. Our approach is illustrated in Figure 7.1.

3.1.2 Structure of the paper

This paper is organized as follows. Data acquisition and preprocessing is presented in Section 3.2, followed up by how we perform motion detection in 3D. In Section 3.3 we describe the concept of motion primitives, and how they are represented compactly by introducing motion context. Furthermore, we show how the motion context can be transformed into a view-invariant representation using spherical harmonic basis functions, yielding a harmonic motion context representation. In Section 3.4 we describe the classification of motion primitives, and how we perform gesture recognition by introducing a probabilistic edit distance classifier. Finally, we present experimental results in Section 7.5 and concluding remarks in Section 7.6.

3.2 Segmentation

3.2.1 Data acquisition and preprocessing

We capture intensity and range data using a SwissRanger SR4000 range camera from MESA Imaging. The camera is based on the Time-of-Flight (ToF) principle and emits radio-frequency modulated (30 MHz) light in the near-infrared spectrum (850 nm), which is backscattered by the scene and detected by a CMOS CCD. The resolution is 176×144 pixels with an active range of 0.3 to 5.0 meters and a field of view of $43.6° \times 34.6°$. The distance accuracy is typically in the order of ± 1 centimeter, depending of the distance range and illumination. Figure 3.2 shows an intensity and a range image of one time instant of a "clap" gesture.

Due to the different reflection properties of materials and the light condition in a captured scene, scattering effects of the active illumination emitted by the camera occurs, hence some noise will be present in the data. To deal with these noise effects we have applied a number of preprocessing techniques proposed in [31]. The preprocessing consists of the following steps: *smoothing of the distance images with a distance-adaptive median filter*, *thresholding on the amplitude values*, and *edge point removal*. Especially, the removal of edge points is of high importance in the case of gesture recognition. Edge points arise in the case where reflected light from the foreground and the background hits the same pixel simultaneously. The camera sensor is controlled as a so-called 1-tap sensor. This means that in order to obtain distance information, four consecutive exposures have to be

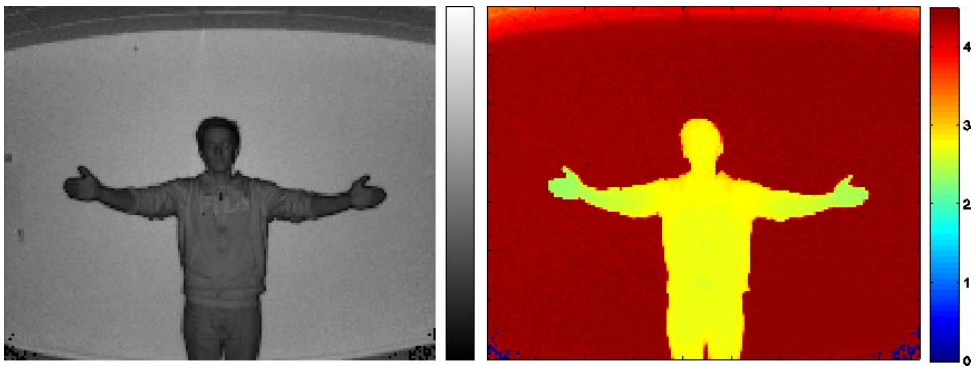


Figure 3.2: An intensity and a range image produced by the SwissRanger SR4000 camera, where the pixel values correspond to a distance in the range image.

performed [22]. If the distance a pixel "sees" changes in this time window, the distance calculation is falsified. The measurement returned by the pixel will be a falsified distance somewhere between the foreground and the background. Fast moving objects in the scene may therefore cause errors in the distance calculations. This error is inversely proportional to the frame rate. The problem especially occurs in regions of a scene that contain objects with high velocity and distance gradients; in our case a fast moving arm. The pixel could then see the arm for the first two taps and a wall for the last two. Thus, the edges of the moving arm are poorly defined and lead to incorrect distance measurement. Concretely, when visualizing the range data as a 3D point cloud, the points origin from these regions are "stretching" backwards along the edges of the moving arm. Since we are interested in gesture recognition where a lot of motion is obviously present, the ToF data can easily be corrupted by a significant amount of neighboring edge points.

3.2.2 3D motion detection

We detect movements (of the arms) using a 3D version of optical flow to produce velocity annotated point clouds [32]. Optical flow is the pattern of apparent motion in a visual scene caused by the relative motion between an observer and the scene. The main benefit of optical flow compared to other motion detection techniques, like double differencing [36] which we have used in an earlier versions of this work [8], is that optical flow determines both the amount of motion and its direction in form of velocity vectors. The following description of the motion detection is inspired by [32] and to some extent quoted or paraphrased. However, the full description along with some additional information and comments are included in this section, as this is an important part of our approach and for the convenience of the reader.

The technique computes the optical flow of each image pixel as the distribution of apparent velocity of moving brightness patterns in an image. The flow of a constant brightness profile can be described by the constant velocity vector $\mathbf{v}_{2D} = (v_x, v_y)^T$ as outlined in Equation 3.1.

$$\begin{aligned} I(x,y,t) &= I(x+\delta x, y+\delta y, t+\delta t) \\ &= I(x+v_x\cdot\delta t, y+v_y\cdot\delta t, t+\delta t) \\ \Rightarrow\ & \frac{\partial I}{\partial x}\cdot v_x + \frac{\partial I}{\partial y}\cdot v_y = -\frac{\partial I}{\partial t} \end{aligned} \tag{3.1}$$

Usually, the estimation of optical flow is based on differential methods. They can be classified into global strategies which attempt to minimize a global energy functional [9] and local methods, that optimize some local energy-like expression. A prominent local optical flow algorithm developed by Lucan and Kanade [16] uses the spatial intensity gradient of an image to find matching candidates using a type of Newton-Raphson iteration. They assume the optical flow to be constant within a certain neighborhood, which allows to solve the optical flow constraint equation (Equation 3.1) via least square minimization.

A characteristic of the Lucas-Kanade algorithm, and that of other local optical flow algorithms, is that it does not yield a very high density of flow vectors, i.e. the flow information fades out quickly across motion boundaries and the inner parts of large homogenous areas show little motion. However, its advantage is the comparative robustness in presence of noise. In the case of the data obtained by ToF cameras, with low resolution and often affected by a high amount of noise, this is a very important property. We use a hierarchical implementation of the Lucas-Kanade algorithm [12] which has successfully been applied in [32].

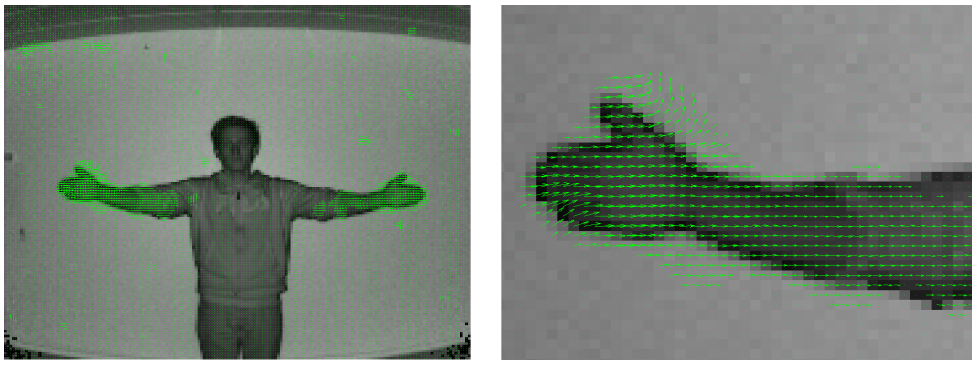


Figure 3.3: An input image overlaid with the estimated 2D optical flow vectors.

The optical flow is computed for each frame $\mathcal{F}_i$ of a sequence of intensity images provided by the SwissRanger camera $(\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_n)$ and based on data from two consecutive frames $(\mathcal{F}_i, \mathcal{F}_{i-1})$. Each pixel of frame $\mathcal{F}_i$ is annotated with a 2D velocity vector $\mathbf{v}_{2D} = (v_x, v_y)^T$ as shown in Figure 3.3, which results in pixel correspondences between frame $\mathcal{F}_i$ and frame $\mathcal{F}_{i-1}$. As a 3D point is available for each pixel these pixel correspondences can be directly transformed into 3D point correspondences $(\mathbf{p}_k^i, \mathbf{p}_l^{i-1})$ which can be used to compute 3D velocities $\mathbf{v}_{3D} = (v_x, v_y, v_z)^T = \mathbf{p}_k^i - \mathbf{p}_l^{i-1}$. Figure 6.4 presents multiple viewpoints of a 3D point cloud of a time instant in a sequence annotated with 3D velocity vectors. In Figure 6.4(left) the moving arms are present in the data, but so is a large amount of noise due to erroneous depth values often produced by the SwissRanger camera. Furthermore, points origin from most of the human body is present due to small

movements or deviations in the distance measurements. These insignificant and erroneous velocity vectors are eliminated to some extent by simple filtering and thresholding as shown in Figure 6.4(right).

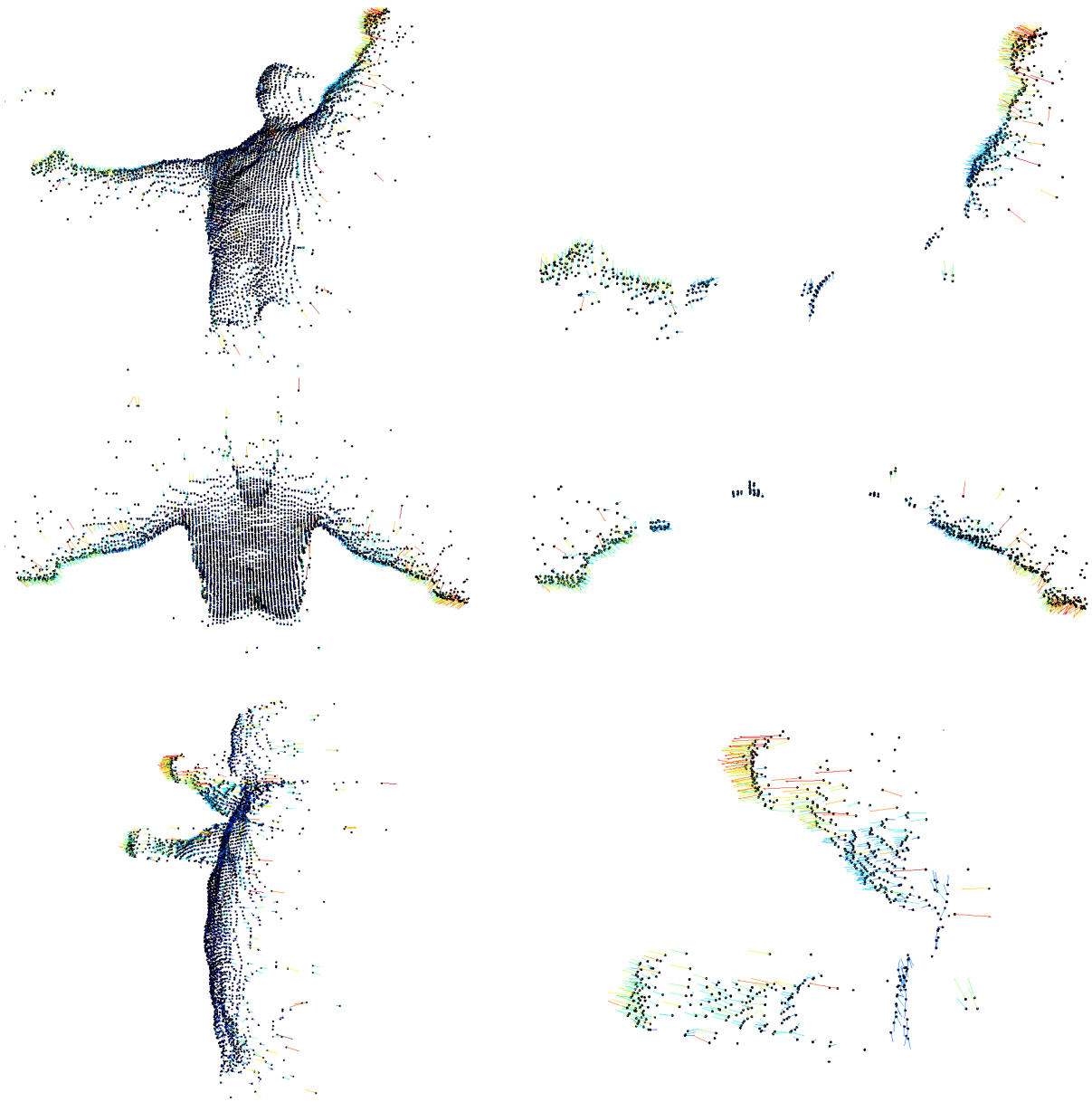


Figure 3.4: Left: Three different viewpoints of a 3D point cloud annotated with 3D velocity vectors. The data has been extracted from a time instant of a "clap" gesture. Right: The same three viewpoints of the velocity annotated point cloud after filtering and thresholding to remove insignificant and erroneous velocity vectors. The points are illustrated with black dots while the velocity vectors are color coded. Blue represents a low velocity while red represents a high velocity. Note: the scale of the sub-figures varies for illustrative purpose.

3.3 Motion primitives

3.3.1 Motion context

After motion detection we are left with a velocity annotated point cloud in 3D, which is represented efficiently using a motion oriented version of shape context. We call this representation the *motion context*.

A shape context [4] is based on a spherical histogram. This histogram is centered in a reference point and divided linearly into S azimuthal (east-west) bins and T colatitudinal (north-south) bins, while the radial direction is divided into U bins. Figure 3.5 gives an example of the shape context descriptor.

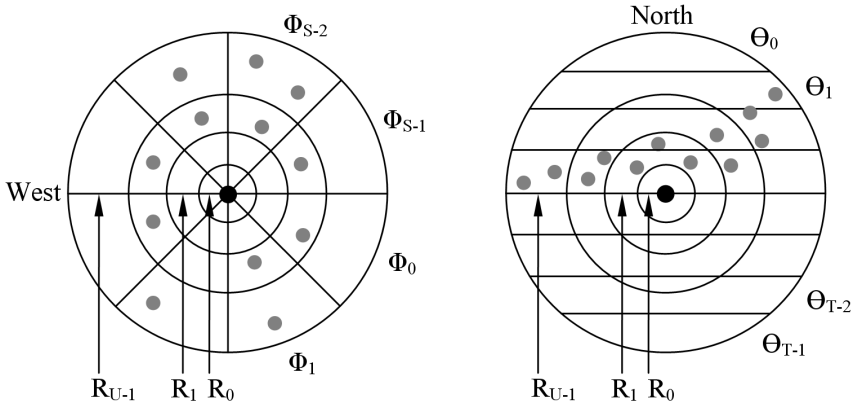


Figure 3.5: A horizontal and a vertical cross-section of a shape context descriptor.

Usually, the value of a bin is given by the number of 3D points falling within that particular bin. However, the motion context extends the regular shape context to represent the velocity annotated point cloud by using both the location of motion, together with the amount of motion and its direction. For each bin in the shape context we accumulate the annotated velocity vectors, of each 3D point falling within that particular bin, into an orientation histogram. Specifically, we introduce a Histogram of Optical Flow (HOF). The idea of HOF is the same as in the Histogram of Oriented Gradients (HOG) used in the popular Scale Invariant Feature Transform (SIFT) [15]. However, we extend this to 3D and in contrast to use gradient vectors, we use velocity vectors. We divide the HOF representation into s azimuthal (east-west) orientation bins and t colatitudinal (north-south) bins, where each bin is weighted by the length of the velocity vectors falling within the bin. This results in a $S \times T \times U \times s \times t$ dimensional feature vector for each frame. The HOF representation and how it is divided into azimuthal and colatitudinal bins is illustrated in Figure 3.6.

In SIFT, partially illumination invariance is imposed by thresholding and normalizing the feature vector. In the same way we impose partially invariance towards the velocity of movements, like in the case where two individuals perform the same gesture at different speed. Hence, the feature vector will have greater emphasis to the location and orientation, while reducing the influence of large velocity values.

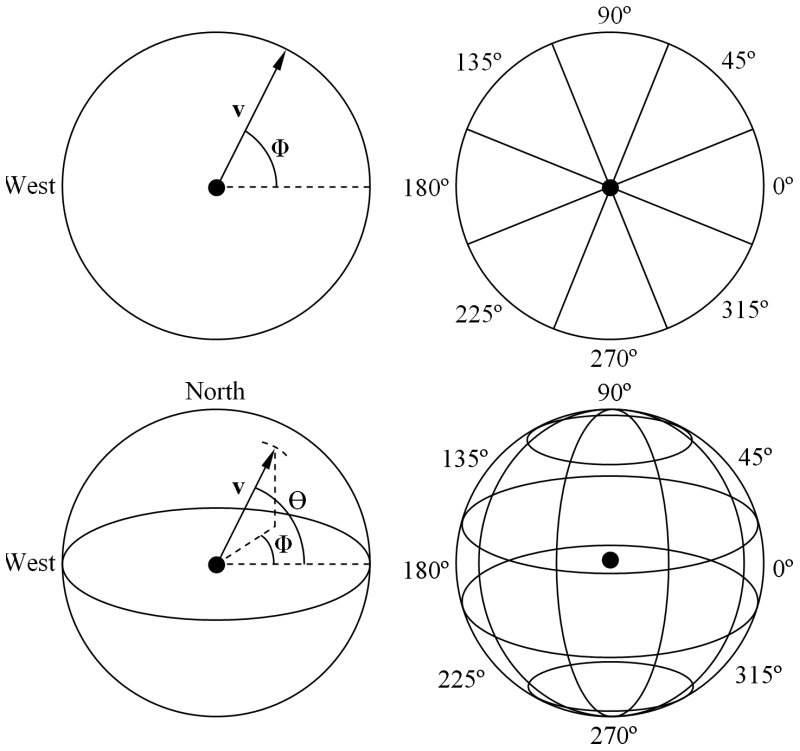


Figure 3.6: The HOF descriptor and how it is divided into 8 azimuthal and 4 colatitudinal bins illustrated by a top-view in 2D and 3D.

3.3.2 View-invariant representation: harmonic motion context

By introducing spherical harmonics we can eliminate one of the two rotational parameters in a shape context descriptor. Similarly, the motion context descriptor can be transformed by using this technique first for each of the HOF descriptors, and thereafter for the entire motion context representation. We eliminate the rotation around the vertical axis, see Figure 3.5 and 3.6, and hereby make our representation invariant to variations in this parameter.

Any given spherical function, i.e. a function $f(\theta, \phi)$ defined on the surface of a sphere parameterized by the colatitudinal and azimuthal variables θ and ϕ, can be decomposed into a weighted sum of spherical harmonics as given by Equation 6.9.

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} A_l^m \, Y_l^m(\theta, \phi) \tag{3.2}$$

The term A_l^m is the weighing coefficient of *degree* m and *order* l, while the complex functions $Y_l^m(\cdot)$ are the actual spherical harmonic functions of *degree* m and *order* l. In Figure 3.7 some examples of higher order spherical harmonic basis functions are illustrated.

The following states the key advantages of the mathematical transform based on the family of orthogonal basis functions in the form of spherical harmonics. The complex function $Y_l^m(\cdot)$ is given by Equation 6.10.

$$Y_l^m(\theta, \phi) = K_l^m \, P_l^{|m|}(\cos\theta) \, e^{jm\phi} \tag{3.3}$$

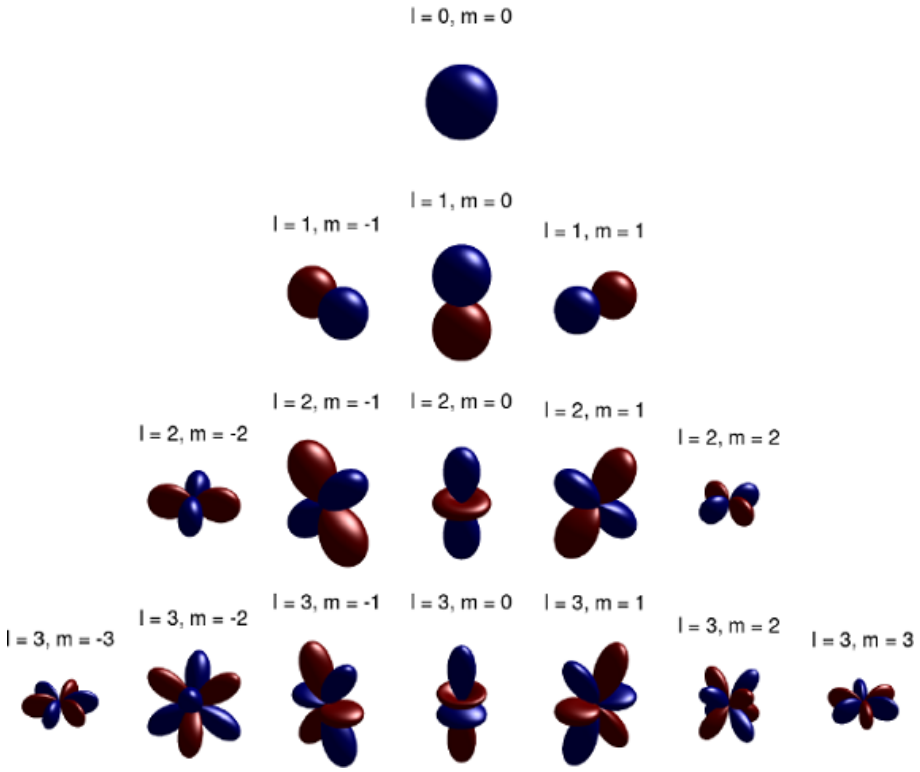


Figure 3.7: Illustration of some higher order spherical harmonic basis functions of degree m and order l. Blue indicates positive values and red negative values.

The term K_l^m is a normalization constant, while the function $P_l^{|m|}(\cdot)$ is the *associated Legendre Polynomial*. The key feature to note from Equation 6.10 is the encoding of the azimuthal variable ϕ. The azimuthal variable solely inflects the *phase* of the spherical harmonic function and has no effect on the *magnitude*. This effectively means that $||A_l^m||$, i.e. the norm of the decomposition coefficients of Equation 6.9 is invariant to parameterization in the variable ϕ.

The actual determination of the spherical harmonic coefficients is based on an inverse summation as given by Equation 6.11, where N is the number of samples $(S \times T)$. The normalization constant $4\pi/N$ originates from the fact, that Equation 6.11 is a discretization of a continuous double integral in spherical coordinates, i.e. $4\pi/N$ is the surface area of each sample on the unit sphere.

$$\left(A_l^m\right)_{f_u} = \frac{4\pi}{N} \sum_{\phi=0}^{2\pi} \sum_{\theta=0}^{\pi} f_u(\theta, \phi)\, Y_l^m(\theta, \phi) \tag{3.4}$$

In a practical application it is not necessary (or possible, as there are infinitely many) to keep all coefficient A_l^m. Contrary, it is assumed the functions f_u (f_u are the spherical functions for each of the given spherical shells $u \in [0; U-1]$) are band-limited, hence it is only necessary to keep coefficient up to some bandwidth $l = B$.

The band-limit assumption effectively means, that each spherical shell is decomposed into $(B+1)^2$ coefficients (i.e., the number of terms in the summation $\sum_{l=0}^{B} \sum_{m=-l}^{l}$ in Equation 6.9). By using the fact, that $||A_l^m|| = ||A_l^{-m}||$ and only saving coefficients

for $m \geq 0$, the number of describing coefficients for each spherical shell is reduced to $(B+1)(B+2)/2$ coefficients (i.e., the number of terms in the summation $\sum_{l=0}^{B}\sum_{m=0}^{l}$). Given the U different spherical shells, the dimensionality of the feature vector becomes $D = U(B+1)(B+2)/2$.

However, since each bin of the spherical motion context representation consists of an embedded spherical function in form of a HOF representation, we first transform each of the inner HOF representations up to some bandwidth B_1, and thereafter we transform the entire motion context up to some bandwidth B_2. Hence, the dimensionality of each transformed HOF representation D_1 and the transformed motion context D_2 becomes $D_1 = (B_1+1)(B_1+2)/2$ and $D_2 = U(B_2+1)(B_2+2)/2$, respectively. When the HOF representations have been transformed, each cell in the motion context consists of an array of spherical harmonic coefficients. This means that the second transformation has to be done with respect to these coefficients, hence the resulting dimensionality of the final feature vector becomes

$$D = D_1 D_2 = U(B_1+1)(B_1+2)(B_2+1)(B_2+2)/4 \tag{3.5}$$

Concretely we set $U = 4$, $B_1 = 4$ and $B_2 = 5$, resulting in 4×315 coefficients (see Figure 3.8).

The spherical motion context histogram is centered in a reference point, which is estimated as the center of gravity of the human body, and the radial division into U bins is made in steps of 25 cm. Furthermore, we set $S = 12$, $T = 6$, $s = 8$ and $t = 4$.

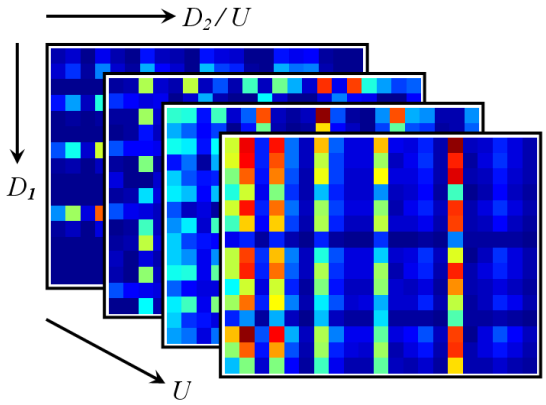


Figure 3.8: An example of a harmonic motion context representation. Where each of the 4 stacked images shows the spherical harmonic coefficients for the 4 radial divisions in the motion context, respectively.

3.4 Classification

The classification is divided into two main tasks: recognition of motion primitives by use of the harmonic motion context descriptors, and recognition of the actual gestures using an ordered sequence of primitives (see Figure 7.1).

3.4.1 Recognition of primitives: correlation

A motion primitive is recognized by matching the current harmonic motion context with a known set, one for each possible primitive. The actual comparison of two harmonic motion contexts is done by the normalized correlation coefficient as given by Equation 3.6. To this end each harmonic motion context is represented as a vector $\mathbf{h_1}$ and $\mathbf{h_2}$ of length n containing the (stacked) spherical harmonic coefficients for a specific primitive at time, t:

$$match(\mathbf{h_1}, \mathbf{h_2}, t) = \frac{n \sum \mathbf{h_1}\mathbf{h_2} - \sum \mathbf{h_1} \sum \mathbf{h_2}}{\sqrt{\left[n \sum (\mathbf{h_1})^2 - (\sum \mathbf{h_1})^2\right]\left[n \sum (\mathbf{h_2})^2 - (\sum \mathbf{h_2})^2\right]}} \tag{3.6}$$

The system is trained by generating a representative set of descriptors for each primitive. A reference descriptor is then estimated as the average of all these descriptors for each class (primitive).

The classification of a sequence can be viewed as a trajectory through the feature space where, at each time-step, the closest primitive (in terms of the normalized correlation coefficient) is found. To reduce noise in this process we introduce a minimum coefficient in order for a primitive to be considered in the first place. Furthermore, to reduce the flickering observed when the trajectory passes through a border region between two primitives we introduce a hysteresis threshold. It favors the primitive recognized in the preceding frame over all other primitives by modifying the individual distances. The classifier hereby obtains a "sticky" effect, which handles a large part of the flickering.

After processing a sequence the output will be a string with the same length as the sequence. An example is illustrated in Equation 3.7. Each letter corresponds to a recognized primitive and Ø corresponds to time instances where no primitives are detected. The string is pruned by first removing 'Ø's, isolated instances, and then all repeated letters, see Equation 3.8. A weight is generated to reflect the number of repeated letters (this is used below).

$$\text{String} = \{Ø, Ø, B, B, B, B, B, E, A, A, F, F, F, F, Ø, D, D, G, G, G, G, Ø\} \tag{3.7}$$

$$\text{String} = \{B, A, F, D, G\} \tag{3.8}$$

$$\text{Weights} = \{5, 2, 4, 2, 4\} \tag{3.9}$$

3.4.2 Recognition of gestures: probabilistic edit distance

The result of recognizing the primitives is a string of letters referring to the known primitives. During a training phase a string representation of each gesture to be recognized is learned. The task is now to compare each of the learned gestures (strings) with the detected string. Since the learned strings and the detected string (possibly including errors!) will in general not have the same length, the standard pattern recognition methods will not suffice. We therefore apply the Edit Distance method [14], which can handle matching of strings of different lengths.

The edit distance is a well known method for comparing words or text strings, e.g., for spell-checking and plagiarism detection. It operates by measuring the distance between two strings in terms of the number of operations needed in order to transform one to the other. There are three possible operations: *insert* a letter from the other string, *delete* a letter, and *exchange* a letter by one from the other string. Whenever one of these operations is required in order to make the strings more similar, the score or distance is increased by one. The algorithm is illustrated in Figure 3.9 where the strings *motions* and *octane* are compared.

The first step is initialization. The two strings are placed along the sides of the matrix, and increasing numbers are placed along the borders beside the strings. Hereafter the matrix is filled cell by cell by traversing one column at a time. Each cell is given the smallest value of the following four operations:

Insert: The value of the cell above + 1

Delete: The value of the cell to the left + 1

Exchange: The value of the cell up-left + 1

No change: The value of the cell up-left + 0. This is the case when the letters in question in the two stings are the same.

Using these rules the matrix is filled and the value found at the bottom right corner is the edit distance required in order to map one string into the other, i.e., the distance between the two strings. The actual sequence of operations can be found by back-tracing the matrix. Note that often more paths are possible.

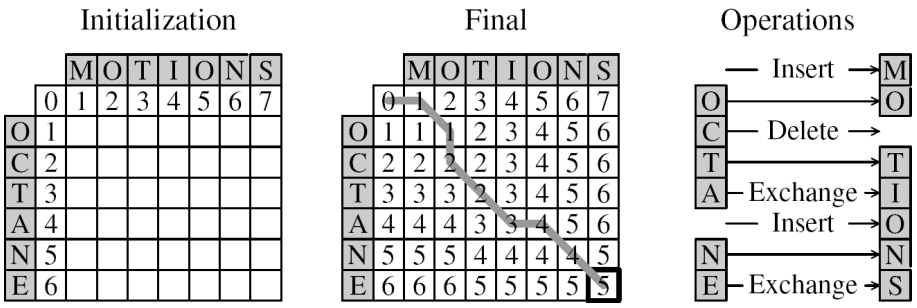


Figure 3.9: Measuring the distance between two strings using edit distance.

When the strings representing the gestures are of different lengths, the method tends to favor the shorter strings. Say we have detected the string $\{B, C, D\}$ and want to classify it as being one of the two gestures: $\#1 = \{J, C, G\}$ and $\#2 = \{A, B, C, D, H\}$. The edit distance from the detected string to the gesture-strings will be two in both cases. However, it seems more likely that the correct interpretation is that the detected string comes from gesture #2 in a situation where the start and end has been corrupted by noise. In fact, 2 out of 3 of the primitives have to be changed for gesture #1 whereas only 2 out of 5 have to be changed for gesture #2. We therefore normalize the edit distance by dividing the output by the length of the gesture-string, yielding 0.67 for gesture #1 and 0.2 for gesture #2, i.e., gesture #2 is recognized.

The edit distance is a deterministic method but by changing the cost of each of the three operations with respect to likelihoods it becomes a probabilistic method[1]. Concretely we apply the weights described above, see Equation 3.9. These to some extent represent the likelihood of a certain primitive being correct. The higher the weight the more likely a primitive will be. We incorporate the weights into the edit distance method by increasing the score by the weight multiplied by β (a scaling factor) whenever a primitive is *deleted* or *exchanged*. The cost of *inserting* remains 1.

The above principle works for situations where the input sequence only contains one gesture (possibly corrupted by noise). In a real scenario, however, we will have sequences which are potentially much longer than a gesture and which might contain more gestures after each other. The gesture recognition problem is therefore formulated as for each gesture to find the substring in the detected string, which has the minimum probabilistic edit distance. The recognized gesture will then be the one of the substrings with the minimum distance. Denoting the start point and length of the substring, s and l, respectively, we recognize the gesture present in the detected string as:

$$\text{Gesture} = \arg\min_{k,s,l} PED(\Lambda, k, s, l) \tag{3.10}$$

where k index the different gestures, Λ is the detected string, and $PED(\cdot)$ is the probabilistic edit distance.

3.5 Test and results

For testing purpose we use a vocabulary consisting of 22 primitives. This is illustrated in Figure 3.10. The criteria for finding the primitives are 1) that they represent characteristic and representative 3D configurations, 2) that their configurations contain a certain amount of motion, and 3) that the primitives are used in the description of as many gestures as possible, i.e., fewer primitives are required. By use of this vocabulary of primitives we describe 4 one- and two-arms gestures: "point right", "raise arm", "clap" and "wave".

We test the system on data recorded of 10 test subjects, each performing the four gestures 2 times from a 0° and ±45° viewpoint with respect to the camera. A total of 160 video sequences have been recorded. Figure 3.11 shows an example of the visual differences that occur when a gesture is performed from these two viewpoints.

To evaluate the view-invariance of the system, the data which is used to train the motion primitives is only from the 0° viewpoint. The overall matching rate is 94.4%. The error distribution can be seen in the confusion matrix in Figure 3.12. In comparison, when only testing on sequences from 0° we obtain a recognition rate of 95.0%.

No significant increase in error can be observed when training and testing on sequences from different viewpoints, i.e., the approach supports view-invariant gesture recognition. The errors observed in both tests are mainly due to personal variations when performing gestures like "point right" and "raise arm". I.e., some tend to raise their arm above the shoulder while pointing while some do not stretch their arm fully when raising their

[1]This is related to the Weighted Edit Distance method, which however has fixed weights.

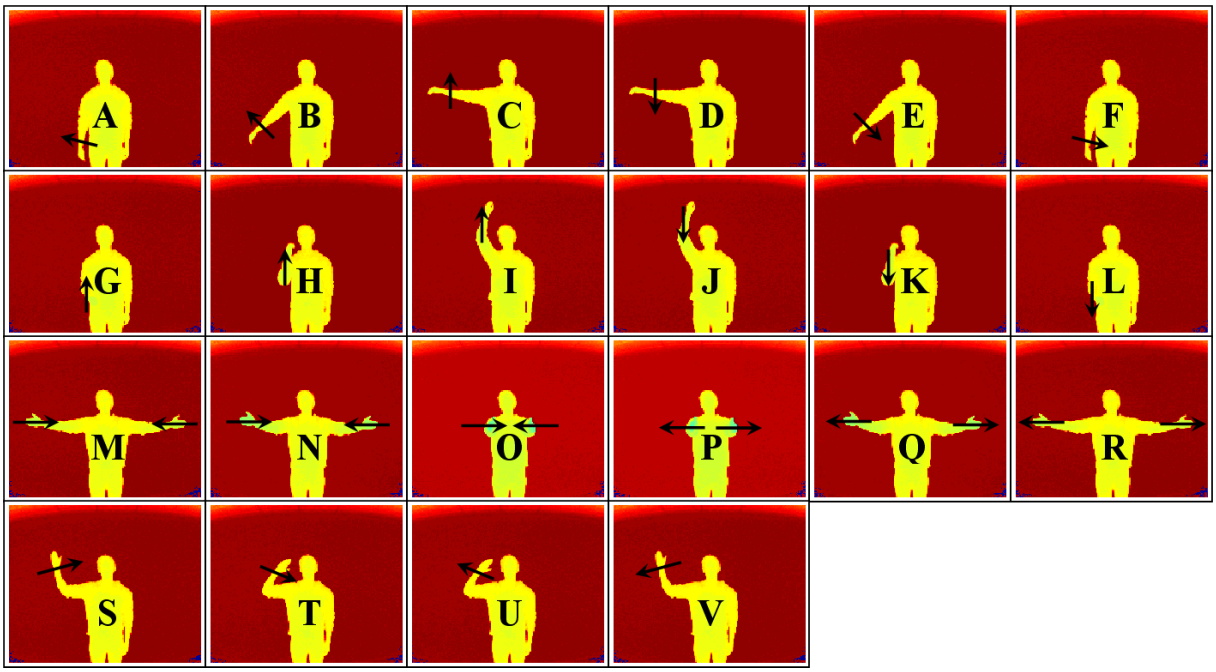


Figure 3.10: The vocabulary consisting of 22 primitives. The primitives are illustrated by range images of the arm configurations and their directions, which are color coded. The color can vary slightly due to error pixels and normalization.

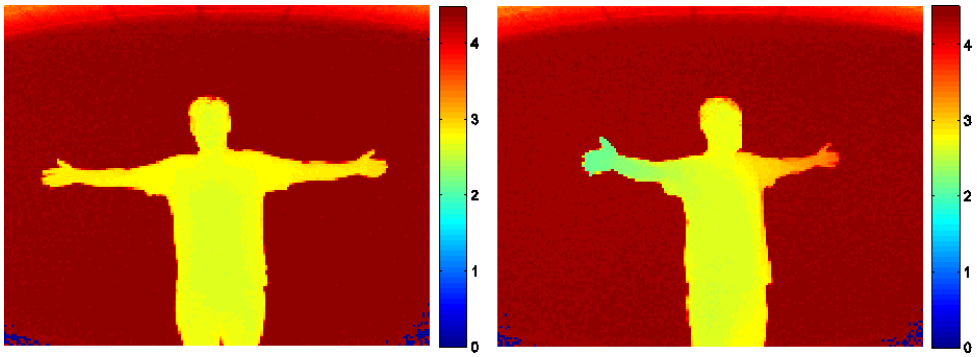


Figure 3.11: Range data examples of a time instance from a video sequence including a person carrying out a "clap" gesture shown from a 0° and +45° camera viewpoint.

	1	2	3	4
1. Point right	90.0	5.0	0.0	5.0
2. Raise arm	7.5	92.5	0.0	0.0
3. Clap	0.0	0.0	95.0	5.0
4. Wave	0.0	0.0	0.0	100.0

Figure 3.12: Test results (given in percentages) for the 4 gestures recorded from a 0° and ±45° viewpoint with respect to the camera.

arm. Another example is in the case of a "clap" gesture, where one of the arms might not be visible or segmented properly due to a too extreme viewpoint when the individual performs this gesture. Hence a "clap" gesture might be classified to be more likely another gesture. In some of these cases the test person turn more than 45° with respect to the camera. As a result most of the arm is hidden behind the body and therefore nearly invisible, hence only very little and poorly defined motion is present.

3.5.1 Unknown start and end time

For each sequence we add "noise" in both the beginning and end of the sequence. By doing so, we introduce the realistic problem of having no clear idea about when a gesture commences and terminates which would be the case in a real situation. To achieve a test scenario that resembles this situation we split the gestures into halves and add one of these half gesture to the beginning and one to the end of each gesture to be processed by the system. The added half gestures are chosen randomly resulting in unknown start and end point of the real gesture.

	1	2	3	4
1. Point right	82.5	7.5	2.5	7.5
2. Raise arm	10.0	87.5	0.0	2.5
3. Clap	0.0	2.5	90.0	7.5
4. Wave	2.5	2.5	7.5	87.5

Figure 3.13: Test results when the start and end time for each gesture are unknown (given in percentages) for the 4 gestures recorded from a 0° and ±45° viewpoint with respect to the camera.

Figure 3.13 shows the confusion matrix for the test results with unknown start and end time. The overall recognition rate for this test is 86.9%, and when only testing on sequences from 0° we obtain a recognition rate of 88.8%. The errors are the same as before but with a few additional errors caused by the unknown start and end time of the gestures. Especially, some "wave" gestures seem to cause falsified classifications. The main part of these errors is caused by confusion between "wave" and "clap" gestures performed at ±45°. If the introduced "noise" include the half of a gesture with movements of the arms in front of the body, like a "clap" gesture, this might lead to such errors as the arms have the same start and end positions for these two gestures.

When comparing our ToF-based 3D gesture recognition approach to 2D methods, the main advantages of our approach are that, by applying 3D data, it is able to build a more descriptive representation in comparison to projected 2D data. This also enables view-invariant representation and recognition. In contrast to other view-invariant methods, which rely on multiple calibrated and synchronized cameras followed up by an exhaustive training phase for multiple viewpoints, our approach is able to recognize gestures by using only one ToF sensor (one viewpoint). Furthermore, we are able to handle unknown gesture commencement and termination, along with variation in gesture speed. The 10 test subjects perform gestures at variable execution time, due to how each individual

perform a certain gesture. Our approach is robust in term of gesture speed variation, since the edit distance metric only needs a few correct matches of each primitive, and can handle missing primitive instances, to correctly classify a given gesture. However, it should be noted that more correct primitive matches strengthens the metric due to the assigned probabilities. In comparison to our previous studies [8], the new motion detection carries more information, and the enhanced view-invariant representation (motion context) is more descriptive and distinctive. The results document this by an improvement in the overall recognition rate.

3.6 Conclusion

The contributions of this paper are twofold. Firstly, motion is detected by 2D optical flow estimated in the intensity image but extended to 3D using the depth information acquired from only one viewpoint by a range camera. We show how gestures can be represented efficiently using motion context, and how gesture recognition can be made view-invariant through the use of 3D data and transforming a motion context representation using spherical harmonics. Secondly, for the gesture recognition system we also address the problem of not knowing when a gesture commences and terminates. This is done by recognizing a gesture *not* through a trajectory based approach, but by representing a gesture as a sequence of discrete primitives, and applying a probabilistic edit distance classifier to identify a given gesture.

The presented approach is trained on gestures from 0° viewpoint and tested on gestures seen from both 0° and ±45° viewpoints. The overall recognition rate is 94.4% with known start and end time of gestures, and 86.9% when the start and end time are unknown. These results state the robustness and view-invariance of the system.

A noticeable extension to the current state of this work would be to develop an automatic primitive selection scheme for the training phase. E.g. a clustering technique could be interesting to investigate further for this purpose.

Acknowledgements

This work is partially funded by the MoPrim and the BigBrother projects (Danish National Research Councils - FTP) and partially by the HERMES project (FP6 IST-027110).

References

[1] R. Koch A. Kolb, E. Barth and R. Larsen. Time-of-Flight Sensors in Computer Graphics. In *Eurographics 2009 - State of the Art Reports*, Munich, Germany, March 2009.

[2] M. Ahmad and S.-W. Lee. HMM-based Human Action Recognition using Multiview Image Sequences. In *International Conference on Pattern Recognition*, Hong Kong, August 2006.

[3] H.H. Avils-Arriaga and L.E. Sucar. Dynamic Bayesian Networks for Visual Recognition of Dynamic Gestures. In *Journal of Intelligent and Fuzzy Systems*, 12(3-4):243-250, 2002.

[4] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition using Shape Contexts. In *Pattern Analysis and Machine Intelligence*, 24(4):509-522, 2002.

[5] I. Cohen and H. Li. Inference of Human Postures by Classification of 3D Human Body Shape. In *Workshop on Analysis and Modeling of Faces and Gestures*, Nice, France, October 2003.

[6] R. Larsen D. Hansen and F. Lauze. Improving Face Detection with TOF Cameras. In *International Symposium on Signals, Circuits and Systems*, Iasi, Romania, July 2007.

[7] R. Ronfard D. Weinland and E. Boyer. Free Viewpoint Action Recognition using Motion History Volumes. In *Computer Vision and Image Understanding*, 104(2):249-257, 2006.

[8] M. Holte, T.B. Moeslund, and P. Fihl. Fusion of Range and Intensity Information for View Invariant Gesture Recognition. In *Workshop on Time-of-Flight based Computer Vision*, Anchorage, Alaska, June 2008.

[9] B. Horn and B. Schunck. Determining Optical Flow. In *Artificial Intelligence*, 17:185-203, August 19810.

[10] F. Roca J. Gonzalez, J. Varona and J. Villanueva. aSpaces: Action Spaces for Recognition and Synthesis of Human Actions. In *International Workshop on Articulated Motion and Deformable Objects*, Palma de Mallorca, Spain, November 2002.

[11] O. Jenkins and M. Mataric. Deriving Action and Behavior Primitives from Human Motion Data. In *International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, September 2002.

[12] S. Khan. LUMS School of Science and Engineering Lahore, Pakistan. http://www.cs.ucf.edu/∼khan/.

[13] R. Larsen and B. Lading. Multiple Geodesic Distance Based Registration of Surfaces Applied to Facial Expression Data. In *International Symposium on Signals, Circuits and Systems*, Iasi, Romania, July 2007.

[14] V. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. In *Soviet Physics Doklady*, 10(8):707-710, 1966.

[15] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. In *International Journal of Computer Vision*, 60(2):91-110, November 2004.

[16] B. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of Imaging Understanding Workshop*, Washington DC, USA, April 1981.

[17] T. Martinetz M. Haker, M. Bohme and E. Barth. Geometric Invariants for Facial Feature Tracking with 3D TOF Cameras. In *International Symposium on Signals, Circuits andSystems*, Iasi, Romania, July 2007.

[18] Switzerland. http://www.mesa-imaging.ch MESA Imaging. Technoparkstrasse 1, 8005 Zuerich.

[19] S. Mitra and T. Acharya. Gesture Recognition: A Survey. In *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 37(3):311-324, 2007.

[20] T. Moeslund, A. Hilton, and V. Kruger. A Survey of Advances in Vision-Based Human Motion Capture and Analysis. In *Computer Vision and Image Understanding*, 104(2):90-126, 2006.

[21] E. Rodner O. Kahler and J. Denzler. On Fusion of Range and Intensity Information Using Graph-Cut for Planar Patch Segmentation. In *Dynamic 3D Imaging Workshop*, Heidelberg, Germany, September 2007.

[22] T. Oggier, M. Stamm, M. Schweizer, and J. Pedersen. User Manual SwissRanger 2 Rev. b. Version 1.02, March 2005.

[23] PMD Technologies. Am Eichenhang 50, D-57076 Siegen, Germany. http://www.pmdtec.com.

[24] A. Prusak, I. Schiller, O. Melnychuk, R. Koch, and H. Roth. Pose Estimation and Map Building with a PMD-Camera for Robot Navigation. In *Dynamic 3D Imaging Workshop*, Heidelberg, Germany, September 2007.

[25] C. Pal R. Messing and H. Kautz. Activity Recognition using the Velocity Histories of Tracked Keypoints. In *International Conference on Computer Vision*, Kyoto, Japan, September 2009.

[26] P.K. Reddy, D. Grest, and V. Kruger. Human Action Recognition in Table-top Scenarios: An HMM-based Analysis to Optimize the Performace. In *Computer Analysis of Images and Patterns*, Vienna, Austria, August 2007.

[27] Q. Shi, L. Wang, L. Cheng, and A. Smola. Discriminative Human Action Segmentation and Recognition using Semi-Markov Model. In *Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008.

[28] V. Kellokumpu S.N. Vitaladevuni and L.S. Davis. Action Recognition using Ballistic Dynamics. In *Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008.

[29] S. Soutschek, J. Penne, J. Hornegger, and J. Kornhuber. 3-D Gesture-Based Scene Navigation in Medical Imaging Applications using Time-of-Flight Cameras. In *Workshop on Time-of-Flight based Computer Vision*, Anchorage, Alaska, June 2008.

[30] R. Souvenir and J. Babbs. Learning the Viewpoint Manifold for Action Recognition. In *Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008.

[31] A. Swadsba, B. Liu, J. Penne, O. Jesorsky, and R. Kompe. A Comprehensive System for 3D Modeling from Range Images Acquired from a 3D ToF Sensor. In *International Conference on Computer Vision Systems*, Bielefeld, Germany, March 2007.

[32] A. Swadzba, N. Beuter, J. Schmidt, and G. Sagerer. Tracking Objects in 6D for Reconstructing Static Scenes. In *Workshop on Time-of-Flight based Computer Vision*, Anchorage, Alaska, June 2008.

[33] 3DV Systems. 2nd Carmel St. Industrial Park Building 1, 20692 Yokneam, Israel. http://www.3dvsystems.com.

[34] C. Thurau and V. Hlavac. Pose Primitive Based Human Action Recognition in Videos or Still Images. In *Computer Vision and Pattern Recognition*, Anchorage, Alaska, June 2008.

[35] W.H.A. Wang and C.L. Tung. Dynamic Hand Gesture Recognition using Hierarchical Dynamic Bayesian Networks through Low-level Image Processing. In *International Conference on Machine Learning and Cybernetics*, Kunming, China, July 2008.

[36] M. Minoh Y. Kameda and K. Ikeda. Three Dimensional Motion Estimation of a Human Body Using a Difference Image Sequence. In *Asian Conference on Computer Vision*, Singapore, December 1995.

[37] A. Yilmaz and M. Shah. Actions Sketch: A Novel Action Representation. In *Computer Vision and Pattern Recognition*, San Diego, CA, USA, June 2005.

Chapter 4

2D Human Action Recognition

This chapter consists of the paper "Selective Spatio-Temporal Interest Points" [A]. The paper presents a spatio-temporal interest point detector for human action recognition in complex scenes, which especially is robust to camera motion and background clutter, where other detectors fails. The Detector separates the space and time domain to suppress background interest points spatially and impose temporal constraints in a second step. Reference [B] describes intermediate work resulting in the final outcome in [A].

References

A. B. Chakraborty, M.B. Holte, T.B. Moeslund and J. Gonzàlez. Selective Spatio-Temporal Interest Points. Accepted for publication in *Computer Vision and Image Understanding, Elsevier*, doi:10.1016/j.cviu.2011.09.010, September 2011.

B. B. Chakraborty, M.B. Holte, T.B. Moeslund and J. Gonzàlez. A Selective Spatio-Temporal Interest Point Detector for Human Action Recognition in Complex Scenes. In *IEEE International Conference on Computer Vision, Barcelona, Spain*, November 2011.

Selective spatio-temporal interest points

B. Chakraborty, M.B. Holte, T.B. Moeslund and J. Gonzàlez

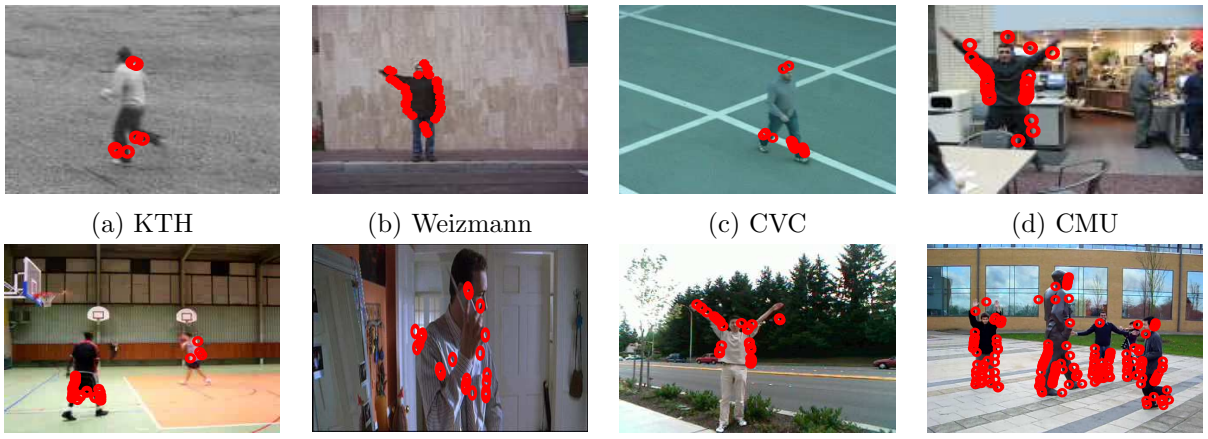
Abstract

Recent progress in the field of human action recognition points towards the use of Spatio-Temporal Interest Points (STIPs) for local descriptor-based recognition strategies. In this paper, we present a novel approach for robust and selective STIP detection, by applying surround suppression combined with local and temporal constraints. This new method is significantly different from existing STIP detection techniques and improves the performance by detecting more repeatable, stable and distinctive STIPs for human actors, while suppressing unwanted background STIPs. For action representation we use a bag-of-video words (BoV) model of local N-jet features to build a vocabulary of visual-words. To this end, we introduce a novel vocabulary building strategy by combining spatial pyramid and vocabulary compression techniques, resulting in improved performance and efficiency. Action class specific Support Vector Machine (SVM) classifiers are trained for categorization of human actions. A comprehensive set of experiments on popular benchmark datasets (KTH and Weizmann), more challenging datasets of complex scenes with background clutter and camera motion (CVC and CMU), movie and YouTube video clips (Hollywood 2 and YouTube), and complex scenes with multiple actors (MSR I and Multi-KTH), validates our approach and show state-of-the-art performance. Due to the unavailability of ground truth action annotation data for the Multi-KTH dataset, we introduce an actor specific spatio-temporal clustering of STIPs to address the problem of automatic action annotation of multiple simultaneous actors. Additionally, we perform cross-data action recognition by training on source datasets (KTH and Weizmann) and testing on completely different and more challenging target datasets (CVC, CMU, MSR I and Multi-KTH). This documents the robustness of our proposed approach in the realistic scenario, using separate training and test datasets.

4.1 Introduction

4.1.1 Human action recognition

In this paper, we address the task of human action recognition in complex scenes in diverse and realistic settings (background clutter, camera motion, occlusions and illumination variations). During the last decade action recognition has been an important topic in the "looking at people" domain [47, 52, 70]. A large number of methods for human action recognition have been proposed, stretching from human model and trajectory-based methods towards holistic and local descriptor-based methods.



(a) KTH (b) Weizmann (c) CVC (d) CMU

(a) YouTube (b) Hollywood 2 (c) MSR I (d) Multi-KTH

Figure 4.1: Example images with superimposed STIPs from the eight action datasets applied for evaluation of our approach: KTH, Weizmann, CVC, CMU, YouTube, Hollywood 2, MSR I and Multi-KTH. The examples give an indication of the described challenges and differences in the datasets: simple scenes (KTH and Weizmann), semi-complex (CVC), and scenes of high complexity (CMU, YouTube, Hollywood 2, MSR I and Multi-KTH).

Most of these previous approaches for human action recognition are constrained to well-controlled environments. Among the proposed action recognition techniques, one type of approach uses motion trajectories to represent actions and it requires target tracking [2, 49]. However, due to the difficulty in building robust object tracker only limited success has been achieved. Another type of approach uses sequences of silhouettes or body contours to model actions [47, 15] and it requires background subtraction. Boiman and Irani [5] extract densely sampled local video patches for detecting irregular actions in videos with simple background. Rodriguez et al. [54] designed a novel method to analyze the filtering responses of different actions. This approach has difficulties in aligning non-repetitive actions in complex scenes. Moreover, some researchers model the configuration of the human body and its evolution in the time domain [23, 79], and others solely perform action recognition from still images by computing pose primitives [65, 75].

The research trend in the field of action recognition has, recently, led to more robust techniques [6, 25, 28, 37, 43, 48, 57, 73, 74, 76], which to some extent are applicable

for action recognition in complex scenes. Action recognition in complex scenes is an extremely difficult task, due to several challenges, like background clutter, camera motion, occlusions and illumination variations. To address these challenges, several methods, like tree-based template matching [25], tensor canonical correlation [28], prototype based action matching [37], a hierarchical approach [48], incremental discriminant analysis of canonical correlation [73], latent pose estimation [74] and generalized Hough transform [76] have been proposed. Most of these methods are very complex and require preprocessing, like segmentation, tree data structure building, target tracking, background subtraction or a human body model. Other methods [8, 12, 18, 17, 21, 26, 31, 36, 41, 39, 40, 45, 50, 53, 55, 56, 61, 64, 67, 68, 69, 77] for action recognition in complex scenes, which demand less or no preprocessing, apply STIP detectors and local descriptors to characterize and encode the video data, and thereby perform action classification.

4.1.2 Spatio-temporal interest points

The extraction of appropriate features is critical to action recognition. Ideally, visual features are able to handle the following challenges for robust performance: (i) scale, rotation and viewpoint variations of the camera, (ii) performance speed variations for different people, (iii) different anthropometry of the actors and their movement style variations, and (iv) cluttered backgrounds and camera motion. The ultimate goal is to be able to perform reliable action recognition applicable for video indexing and search, intelligent human computer interaction, video surveillance, automatic activity analysis and behavior understanding. Recently, the use of STIPs has received increasing interest for local descriptor-based action recognition strategies. STIP-based methods avoid the temporal alignment problem, are exceptionally invariant to geometric transformations, and therefore distorted less by changes in scale, rotation and viewpoint than image data. Features are locally detected, thus inherently robust to occlusion and do not suffer from conventional figure-ground segmentation problems (imprecise segmentation, object splitting and merging etc.). Additionally, partial robustness to illumination variations and background clutter are incorporated.

Laptev and Lindeberg first proposed STIPs for action recognition [34], by introducing a space-time extension of the popular Harris detector [22]. They detect regions having high intensity variation in both space and time as spatio-temporal corners. The STIP detector of [34] usually suffers from sparse STIP detection. Later several other methods for detecting STIPs have been reported [11, 24, 51, 71, 72]. Dollár et al. [11] improved the sparse STIP detector by applying temporal Gabor filters and select regions of high responses. Dense and scale-invariant spatio-temporal interest points were proposed by Willems et al. [71], as a spatio-temporal extension of the Hessian saliency measure, previously applied for object detection [4, 38]. Instead of applying local information for STIP detection Wong and Cipolla [72] propose a global information-based approach. They use global structural information of moving points and select STIPs according to their probability of belonging to the relevant motion. Although promising results have been reported, these methods are quite vulnerable to camera motion and cluttered background, since they detect interest points directly in a spatio-temporal space.

Hence, STIP-based methods have some shortcomings. First of all, (i) STIPs focus on

local spatio-temporal information instead of global motion, thus the detection of STIPs on human actors in complex scenes might fall on cluttered backgrounds, especially if the camera is not fixed. Secondly, (ii) the stability of STIPs varies due to the local properties of the detector, and therefore some STIPs can be unstable and imprecise, as a result they have low repeatability or the local descriptors can become ambiguous. Thirdly, (iii) redundancy can occur in the local descriptors extracted from the surrounding image region of two adjacent STIPs. According to Schmid et al. [58] robust interest points should have high repeatability (geometric stability) and information content (distinctiveness of features). Furthermore, Turcot and Lowe [66] investigate and report that it is better to select a small subset of useful features for recognition problems, than a larger set of unreliable features which represent irrelevant clutter. We address these three shortcomings, by first (i) detecting Spatial Interest Points (SIPs), then (ii) suppressing unwanted background points, and finally (iii) imposing local and (iv) temporal constraints, achieving a set of selective STIPs which are more robust to these challenges.

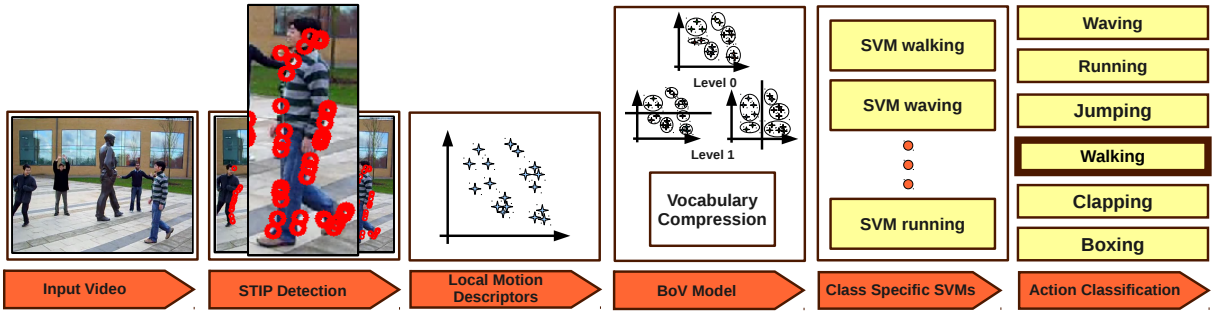


Figure 4.2: A schematic overview of the system structure and data flow pipeline of our approach.

4.1.3 Local descriptors

Several local descriptors have been proposed in the past few years [11, 71, 29, 30, 35, 36, 60]. Local feature descriptors extract shape and motion in the neighborhoods of selected STIPs using image measurements, such as spatial or spatio-temporal image gradients or optical flow. Laptev et al. [36] introduced a combined descriptor to characterize local motion and appearance by computing histograms of spatial gradient (HOG) and optic flow (HOF) accumulated in space-time neighborhoods of detected interest points. Willems et al. [71] proposed the Extended SURF (ESURF) descriptor, which extends the image SURF descriptor [3] to videos. The authors divide 3D patches into cells, where each cell is represented by a vector of weighted sums of uniformly sampled responses of the Haar-wavelets along the three axes. Dollár et al. [11] proposed a descriptor along with their detector. The authors concatenate the gradients computed for each pixel in the neighborhood into a single vector and apply Principal Component Analysis (PCA) to project the feature vector onto a low dimensional space. Compared to the HOG-HOF descriptor proposed by Laptev et al. [36], it does not distinguish the appearance and motion features. The 3D-SIFT descriptor was developed by Scovanner et al. [60]. This descriptor is similar to the Scale Invariant Feature Transformation (SIFT) descriptor [42], except

that it is extended to video sequences by computing the gradient direction for each pixel spatio-temporally in three-dimensions. Another extension of the popular SIFT descriptor was proposed by Kläser et al. [29]. It is based on histograms of 3D gradient orientations, where gradients are computed using an integral video representation. Another popular descriptor is the N-jets [30, 33]. An N-jet is the set of partial derivatives of a function up to order N, and is usually computed from a scale-space representation. The N-jets is an inherently strong local motion descriptor, where the two first levels implicitly represent velocity and acceleration.

4.1.4 Vocabulary building strategies

Bag-of-video words (BoV) models have become popular for generic action recognition [11, 34, 39, 40, 72, 78], whereas other techniques based on co-occurrence of STIP based motion features are also used [46]. The basic BoV model computes and quantizes the feature vectors, extracted at the detected STIPs in the video, into video-words. Finally, the entire video sequence is represented by a statistical distribution of those video-words. For classification, discriminative learning models such as SVM [11] and generative models, e.g. pLSA [72], have achieved excellent performance for action recognition. Since the BoV model does not provide a spatio-temporal distribution of features, the spatial correlogram and spatio-temporal pyramid matching are applied [40, 45] to capture the spatio-temporal relationship between local features. Additionally, vocabulary compression techniques are used to reduce the final feature space [39, 40]. We introduce a novel vocabulary building strategy by first applying a spatial pyramid and then compress the vocabulary at each pyramid level, achieving a compact and efficient pyramid representation of actions. This is different from [40], where first a vocabulary is computed, then it is compressed, and finally a spatial correlogram and a spatio-temporal pyramid are applied.

4.1.5 Complex scenes

While reliable human action recognition in simple scenes (KTH [59] and Weizmann [19]) has been achieved [8, 17, 26, 28, 37, 73], the task remains unsolved for complex scenes. These datasets have been recorded in well-controlled environments with clean or simple background, controlled lighting conditions, and no camera motion nor occlusions. In contrast, Real world human actions are often recorded in scenes of high complexity, with cluttered background, illumination variations, camera motion and occluded bodies. Hence, these datasets do not correspond very well to real world scenarios. The mentioned properties make action recognition in complex scenes much more challenging. New datasets for the purpose of evaluation of action recognition algorithms in complex and semi-complex scenes have therefore been produced (CMU [27], CVC [1], YouTube [39], Hollywood 2 [45], MSR I [78] and Multi-KTH [67]). We utilize all these datasets for evaluation of our approach (see Figure 4.1).

4.1.6 Cross-data evaluation

Conventional approaches usually build a classifier from labeled examples and assume the test samples are generated from the same distribution, which is rarely the case in realistic scenarios. In contrast, cross-data evaluation is highly necessary for commercial systems, where the classifier is trained on a specific dataset during a learning phase and then set up for operation in the field. Additionally, it also prevents the algorithm to benefit from the internal data correlation during the evaluation. Cross-data evaluation is more challenging, since the two dataset have usually been recorded in two different occasions. Only a few authors have recently reported cross-data evaluation [8, 17, 67]. The problem is related to transfer learning known from machine learning, which attempts to develop methods to transfer knowledge learned in one or more source tasks and use it to improve learning in a related target task [63, 10]. We conduct a comprehensive set of cross-data experiments to carry out a more realistic evaluation of our approach.

4.1.7 Our approach and contributions

In this work we follow the recent progress and employ a STIP and local descriptor-based recognition strategy. A schematic overview of our approach is outlined in Figure 4.2. (1) We introduce a novel approach for selective STIP detection, by applying surround suppression combined with local and temporal constraints, achieving robustness to camera motion and background clutter. For action representation we use a BoV model of local N-jet features, extracted at the detected STIPs, to build a vocabulary of visual-words. (2) To this end, we introduce a novel vocabulary building strategy by combining (i) a pyramid structure to capture spatial information, and (ii) vocabulary compression to reduce the dimensionality of the feature space, resulting in improved performance and efficiency. Action class-specific SVM classifiers are trained and applied for categorization of natural human actions. (3) We evaluate our approach on both popular benchmark datasets (KTH and Weizmann), more challenging datasets (CVC, CMU), movie and YouTube video clips (Hollywood 2 and YouTube) and perform an exhaustive cross-data evaluation, trained on source dataset (KTH and Weizmann) and tested on more challenging target datasets (CVC, CMU, MSR I and Multi-KTH). Due to the unavailability of ground truth action annotation data for the Multi-KTH dataset, we introduce an actor specific spatio-temporal clustering of STIPs to address the problem of automatic action annotation of multiple simultaneous actors. To observe the performance our automatic STIP clustering-based annotation, we manually annotate the ground truth actions and compare the action recognition accuracies. Finally, we compare our approach to the most popular action recognition techniques and show beyond state-of-the-art performance.

4.1.8 Paper structure

The remainder of the paper is organized as follows. We describe our STIP detector and local descriptor-based action representation in section 4.2. Section 6.4 outlines our vocabulary building strategy and narrates the applied classifier for action categorization. Experimental results and comparisons, along with our technique for spatio-temporal clus-

tering of STIPs for automatic action annotation of Multi-KTH, are reported in section 7.5, followed up by concluding remarks in section 6.6.

4.2 Selective spatio-temporal interest points

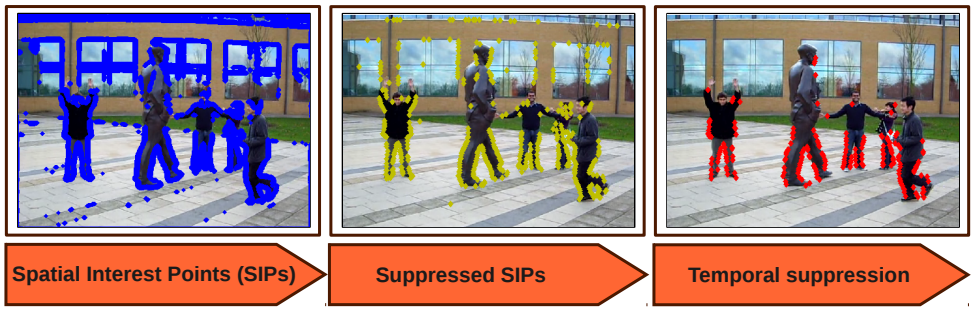
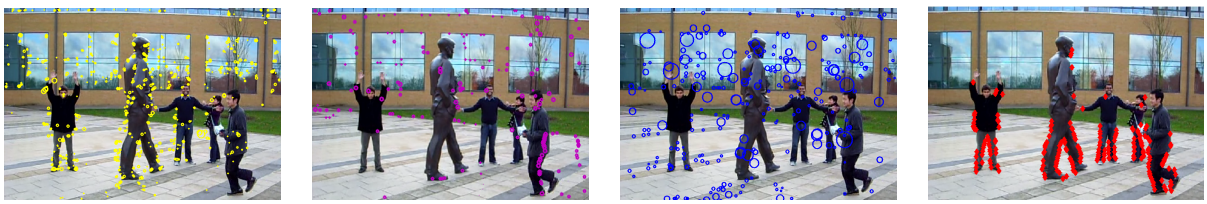


Figure 4.3: A schematic overview of the spatio-temporal interest point detection module and the associated data flow pipeline.

4.2.1 Detection of spatial interest points.

Existing STIP detectors [11, 24, 34, 71, 72] are vulnerable to camera motion and moving background in videos, and therefore detect unwanted STIPs in the background (see Figure 4.4). Cao et al. [8] have recently reported, that of all the STIPs detected by Laptev's STIP detector [34], only about 18% correspond to the three actions performed by the actors in the MSR I dataset [78], while the rest of the STIPs (82%) belong to the background. To overcome this problem, we first detect the spatial interest points (SIPs), then perform background suppression and impose local and temporal constraints (see Figure 4.3). We apply the basic Harris corner detector [22] and compute the first set of interest points with corner strength C_σ, where σ is the spatial scale. Apart from the detected SIPs on the human actors, the obtained spatial corners C_σ contain a significant amount of unwanted background SIPs (see Figure 4.3).



(a) (b) (c) (d)

Figure 4.4: STIP detection results for the Multi-KTH dataset. (a) Laptev et al. [34], (b) Dollár et al. [11], (c) Willems et al. [71] and (d) Our approach. Due to background clutter and camera motion (a), (b) and (c) detect quite a large number of STIPs in the background compared to our approach.

4.2.2 Suppressing background interest points

The main idea of our spatial interest point suppression originates in the fact that most corner points detected in the background texture or on non-human objects follow some particular geometric pattern, while those on humans do not have this property. For suppression we use a surround suppression mask (SSM) for each interest point, taking the current point under evaluation as the center of the mask. We then estimate the influence of all surrounding points of the mask on the central point, and accordingly, a suppression decision is taken. The idea is motivated by Grigorescu et al. [20], where surround suppression is used for texture edges to improve object contour and boundary detection in natural scenes. The similar concept of surround suppression based on center surround saliency measure is been adopted in tracking [14], spatio-temporal saliency algorithm [44] and detection of suspicious coincidences in visual recognition [16]. We implement surround suppression by computing an inhibition term for each point of C_σ. For this purpose we introduce a gradient weighting factor $\triangle_{\Theta,\sigma}(x, y, x-u, y-v)$, which is defined as:

$$\begin{aligned} &\triangle_{\Theta,\sigma}(x, y, x-u, y-v) = \\ &|\cos(\Theta_\sigma(x, y) - \Theta_\sigma(x-u, y-v))| \end{aligned} \tag{4.1}$$

where $\Theta_\sigma(x, y)$ and $\Theta_\sigma(x-u, y-v)$ are the gradients at point (x, y) and $(x-u, y-v)$, respectively; u and v define the horizontal and vertical range of the SSM. If the gradient orientations at point (x, y) and $(x-u, y-v)$ are identical, the weighting factor attains its maximum ($\triangle_{\Theta,\sigma} = 1$), while the value of the factor decreases with the angle difference and reaches a minimum ($\triangle_{\Theta,\sigma} = 0$), when the two gradient orientations are orthogonal. Hence, the surrounding interest points which have the same orientation, as that of (x, y), will have a maximal inhibitory effect.

For each interest point $C_\sigma(x, y)$, we define a suppression term $t_\sigma(x, y)$ as the weighted sum of gradient weights in the suppression surround of that point:

$$\begin{aligned} t_\sigma(x, y) \;\; = \;\; & \iint_\Omega C_\sigma(x-u, y-v) \\ & \times \triangle_{\Theta,\sigma}(x, y, x-u, y-u) du dv \end{aligned} \tag{4.2}$$

where Ω is the image coordinate domain. We now introduce an operator $C_{\alpha,\sigma}(x, y)$, which takes its inputs: the corner magnitude $C_\sigma(x, y)$ and the suppression term $t_\sigma(x, y)$:

$$C_{\alpha,\sigma}(x, y) = H(C_\sigma(x, y) - \alpha t_\sigma(x, y)) \tag{4.3}$$

where $H(z) = z$ when $z \geq 0$ and *zero* for negative z values. The factor α controls the strength of the surround suppression. If no interest points have been detected in the surrounding texture of a given point, the response of the operator retains the original corner magnitude $C_\sigma(x, y)$. However, if a large number of interest points are detected in the surrounding background texture, the suppression term $t_\sigma(x, y)$ will be higher, resulting in a suppression of the current interest point under evaluation.

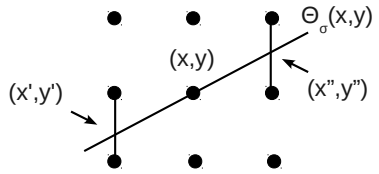


Figure 4.5: Responses at position (x', y') and (x'', y'') along the line passing through (x, y) [20]. Non-maxima suppression retains the value in the central position (x, y), if it is greater than the values at (x', y') and (x'', y'').

4.2.3 Imposing local constraints

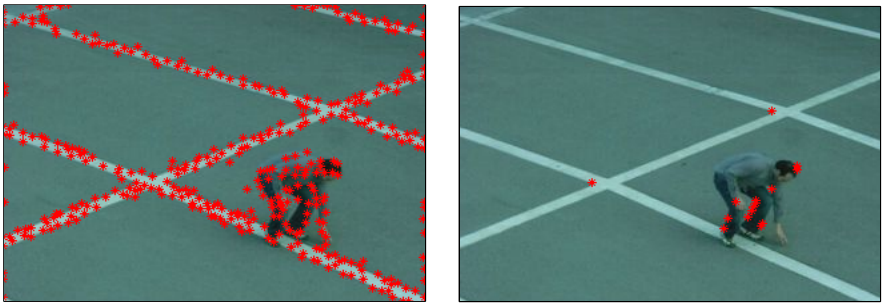
We select a final set of interest points from the surround suppression responses $C_{\alpha,\sigma}$ (Equation 6.4) by applying non-maxima suppression, similar to Grigorescu et al.'s method for suppressing gradients [20]. Non-maxima suppression thins the areas in which $C_{\alpha,\sigma}$ is non-zero to one-pixel wide candidate contours as follows: for each position (x, y), the two responses $C_{\alpha,\sigma}(x', y')$ and $C_{\alpha,\sigma}(x'', y'')$ in adjacent positions (x', y') and (x'', y''), which are intersection points of a line passing through (x, y) with orientation $\Theta_\sigma(x, y)$ and a square defined by the diagonal points of an 8-neighborhood, are computed by linear interpolation (see Figure 4.5). A point is kept, if the response $C_{\alpha,\sigma}(x, y)$ is greater than that of the two adjacent points, i.e., it is a local maximum of the neighborhood. Otherwise its value is set to zero. Figure 4.6 shows an example of the performance of our inhibitive SIP detector. As can be seen in Figure 4.6.b some background SIPs might remain in $C_{\alpha,\sigma}$. However, these static SIPs can be removed by imposing temporal constraints.

4.2.4 Scale adaptive SIPs

Scale selection plays an important role in the detection of spatial interest points. Automatic scale selection can be achieved based on the maximization of normalized derivatives expressed over scale, or by the behavior of entropy or error measures evaluated over scale [7, 38]. Instead of applying an automatic scale selection, as in [32], we apply a multi-scale approach [36] and compute suppressed SIPs in *five* different scales $S_\sigma = \{\frac{\sigma}{4}, \frac{\sigma}{2}, \sigma, 2\sigma, 4\sigma\}$. We follow the idea of scale selection presented by Lindeberg [38] to keep the best set of SIPs obtained for each scale. The best scales are selected by maximizing the normalized differential invariant,

$$\tilde{\kappa}_{norm} = \sigma_0^{2\gamma} L_y L_{xx}. \tag{4.4}$$

where $L = g(\cdot; \sigma_0, \tau_0) \otimes I$, i.e. the image I is convoluted with the Gaussian kernel g; L_y is the first order y derivative and L_{xx} is the second order x derivative of L. Lindeberg [38] report that $\gamma = \frac{7}{8}$ performs well in practice to achieve the maximum value of $(\tilde{\kappa}_{norm})^2$ for spatial interest point detected at multiple scales. After computing the suppressed SIPs in the scale-space in S_σ, we apply this scale selection procedure based on the normalized differential invariant (Equation 6.5), and keep the n best SIPs as our final set of suppressed SIPs.



(a) (b)

Figure 4.6: Performance of our SIP detector with $\alpha = 1.5$. Detected SIPs (a) before suppression and (b) after suppression.

4.2.5 Imposing temporal constraints

After obtaining the final set of spatial interest points we impose temporal constraints to neglect static SIPs. We consider two consecutive frames at a time and remove the common interest points, since static interest points do not contribute any motion information:

$$\mathcal{P}^T_{\alpha,\sigma} = C^T_{\alpha,\sigma} \backslash \{C^T_{\alpha,\sigma} \cap C^{T-1}_{\alpha,\sigma}\} \tag{4.5}$$

where $C^T_{\alpha,\sigma}$ is the set of interest points in the T^{th} frame. To avoid the camera motion we have used an interest point matching algorithm along with a temporal Gabor filter response to remove the static interest points (Equation 4.5). The remaining points are the final set of detected STIPs, which are used to extract local features. The pseudo code for the full STIP detection is described in Algorithm 1. Parallelization can be adopted for speed optimization by parallel computation of the **for** loops in each algorithm (Algorithm 1,3,2,4 and 5).

4.2.6 Local feature descriptors

We use local N-jet features [30] extracted at the detected STIPs. We extract N-jet features of order-2 in five different temporal scales. Consequently, we end up with a 10-dimensional feature vector,

$$\mathcal{F}_{norm}(g(\cdot;\sigma_0,\tau_0) \cdot I) = \{L, \sigma L_x, \sigma L_y, \ldots, \tau^2 L_{tt}\} \tag{4.6}$$

at locally adopted scale level (σ_0, τ_0) for the image sequence I; where $g(\cdot;\sigma_0,\tau_0)$ is the Gaussian kernel at spatio-temporal scale (σ_0, τ_0) and σ_0 is identical to the scale of the STIP detector; $L = g(\cdot;\sigma_0,\tau_0) \otimes I$, i.e. the image I is convoluted with the Gaussian kernel g; L_x is the first order x derivative and L_{xx} is the second order x derivative of L etc.

These features are computed with a fixed spatial scale σ_0 but with five different temporal scales $(\frac{\tau}{4}, \frac{\tau}{2}, \tau, 2\tau, 4\tau)$. We do not increase the order of N-jet, like Laptev et al. [33], since the two first levels represent velocity L_{xt} and acceleration L_{tt} information, while higher order spatial or temporal derivatives are sensitive to noise and do not bring significant additional motion information. The experimental results reported in section 7.5 document our feature selection by showing state-of-the-art performance.

Algorithm 1 STIP detection from an image stack.

Require: An image stack (H × W × N): iS;
(contains all the video frames)
Array containing spatial scales: sA;
Alpha: α;
Mask: m;
Ensure: Detected STIPs: $stip$
$sip = \{\}$; $stip = \{\}$;
$N = size(iS, 3)$; (Total no. of frames)
for $i = 1 \rightarrow N$ **do**
  for $j = 1 \rightarrow size(sA)$ **do**
    $sip \leftarrow sip \cup \{SCD(iS(:,:,i), sA(j), \alpha, m), sA(j)\}$;
  end for
  $stip \leftarrow stip \cup blobDetector(iS(:,:,i), sip)$;
end for
$stip = temporalConstraint(iS, stip)$;
Return($stip$);

Algorithm 2 SCD: Selective STIP detection.

Require: An image (H × W): $image$;
Spatial scale: σ;
Alpha: α;
Mask: $mask$;
Ensure: Detected selective spatial interest points: sip
$cp = harrisCorner(image, \sigma)$;
$cornerPoints = find(cp > 0)$;
$cp = cp(cornerPoints)$;
$\Theta = gradient(image)$;
$sip = \{\}$;
for Each point $(x, y, \sigma) \in cornerPoints$ **do**
  $\triangle_{\Theta_{mask}} = |\cos(\Theta_{mask} - \Theta_{mask_{(x,y)}})|$;
  $t(x, y) = cp_{mask} \otimes \triangle_{\Theta_{mask}}$;
  $cp(x, y) = H(cp_{(x,y)} - \alpha t_{(x,y)})$;
  $(x', y') = round(line(x, x + 1, y, \Theta(x, y)))$;
  $(x'', y'') = round(line(x, x - 1, y, \Theta(x, y)))$;
  if $(cp(x, y) > cp(x', y')) \wedge (cp(x, y) > cp(x'', y''))$ **then**
    $sip \leftarrow sip \cup (x, y, \sigma)$;
  end if
end for
Return(sip);

4.3 Vocabulary building and classification

We apply a BoV model to learn the visual vocabularies of the extracted local motion features. We extend the idea of [39] by introducing pyramid levels in the feature space,

Algorithm 3 blobDetector: Corner strength detection using Gaussian blob.

Require: An image (H × W): im;
Corner points: $corners$;
Ensure: Detected selective spatial interest points based on Gaussian blob strength: $cornerPoints$

$cornerPoints = \{\}$;
for Each point $(X, Y, \sigma) \in corners$ **do**
  $bS = \sigma^{1.75} * L_{y,im}(X, Y) * L_{xx,im}(X, Y)$;
  if $(bS > \tau)$ **then**
    $cornerPoints \leftarrow cornerPoints \cup (X, Y, \sigma)$;
  end if
end for
Return($cornerPoints$);

Algorithm 4 temporalConstraint: Imposed temporal constraint on the selected spatial corner points

Require: An image stack (H × W × N): iS;
Spatial corner points: cp;
Ensure: Detected STIPs: $stip$

for $i = 1 \rightarrow H$ **do**
  for $j = 1 \rightarrow W$ **do**
    $gabor(i, j, :) = gaborFiler1D(iS(i, j, :))$;
  end for
end for
for $i = N \rightarrow 2$ **do**
  $f_1 = iS(:, :, i)$; $f_2 = iS(:, :, i - 1)$;
  $g_1 = gabor(:, :, i)$; $g_2 = gabor(:, :, i - 1)$;
  $im_1 = iS(:, :, i)$; $im_2 = iS(:, :, i - 1)$;
  $cp_{f_1} \leftarrow cp_{f_1} \backslash pointMatch(cp_{f_1}, cp_{f_2}, g_1, g_2, im_1, im_2)$;
end for
Return(cp)

but instead of applying a pyramid at feature level, as in [40], we apply it at STIP level. This makes the problem of grouping the local features much simpler yet robust, since our STIPs are detected in a selective and robust manner. Finally, we apply vocabulary compression, at each pyramid level, to reduce the dimensionality of the feature space (see Figure 4.7).

4.3.1 Pyramid structure

Let I_T be the T^{th} frame of the image sequence I and $P^T_{\alpha,\sigma}$ (Equation 4.5) the set of detected STIPs in this frame. We then quantize this set of STIPs into q levels, $\mathcal{S} = \{s_0, s_1, \ldots, s_{q-1}\}$ [45]. For each of these levels, the STIPs are divided based on center of mass information. Accordingly, we group the motion features into different levels of the pyramid. The

Algorithm 5 pointMatch: Detect the set of matching corner points in two consecutive frames.

Require: Image frames: im_1, im_2;
Corner strengths: cp_1, cp_2;
Gabor strength: g_1, g_2;
Ensure: Detected matching STIPs: mS

```
mP = {};
cornerPoints_1 = find(cp_1 > 0);
cornerPoints_2 = find(cp_2 > 0);
for Each point (x_1, y_1, σ_1) ∈ cornerPoints_1 do
    H = σ_1;
    for Each point (x_2, y_2, σ_2) ∈ cornerPoints_2 do
        similarity = min(cp_1(x_1,y_1),cp_2(x_2,y_2)) / min(cp_1(x_1,y_1),cp_2(x_2,y_2));
        W = σ_2;
        if similarity > τ_sim then
            a_1 = cropRect(im_1, x_1, y_1, H, W);
            a_2 = cropRect(im_2, x_2, y_2, H, W);
            sC = crossCorrelation(a_1, a_2);
            if (sC > τ_corr) ∧ (g_1(x_1, y_1) > τ_gabor) then
                mP ← mP ∪ (x_1, y_1, σ_1);
            end if
        end if
    end for
end for
Return(mS);
```

structure of our 2-level pyramid is illustrated in Figure 6.7. The horizontal division helps to capture the distinguishing characteristics of arm and leg-based actions, whereas the vertical division distinguishes the actions within each of these arm and leg-based action classes.

4.3.2 Vocabulary compression

After dividing the motion features into the described pyramid levels, we create initial vocabularies of a relatively large size (about 400 words). To reduce the final feature dimensionality, we use vocabulary compression, as in [39], but at each level of the pyramid to achieve a compact yet discriminative visual-word representation of actions.

Let A be a discrete random variable which takes the value of a set of action classes $A = \{a_1, a_2, \ldots, a_n\}$, and W_s be a random variable which range over the set of video-words $W_s = \{w_1, w_2, \ldots, w_m\}$ at pyramid level s. Then the information about A captured by W_s can be expressed by the Mutual Information (MI), $I(A, W_s)$. Now, let $\widehat{W_s} = \{\hat{w_1}, \hat{w_2}, \ldots, \hat{w_k}\}$ for $k < m$, be the compressed video-word cluster of W_s. We can measure the loss of quality of the resulting compressed vocabulary $\widehat{W_s}$, as the loss of MI:

$$Q(\widehat{W_s}) = I(A, W_s) - I(A, \widehat{W_s}) \tag{4.7}$$

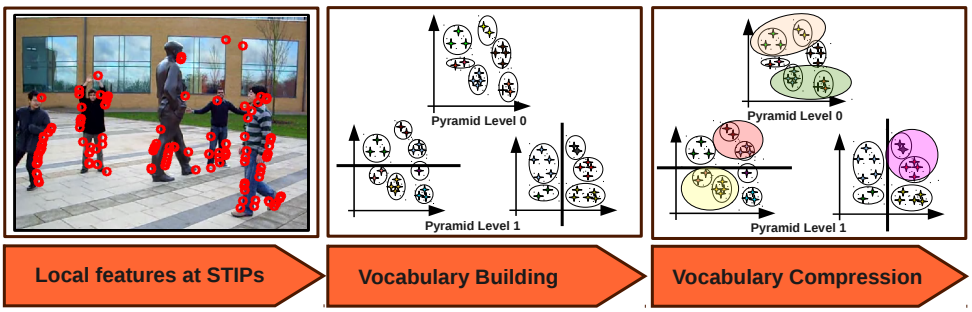


Figure 4.7: A schematic overview of the vocabulary building module and the associated data flow pipeline.

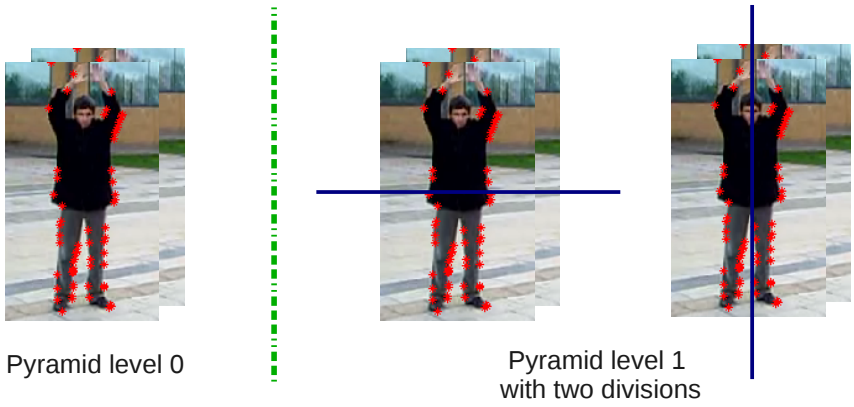


Figure 4.8: Spatial pyramid of level 2.

To find the optimal compression $\widehat{W}_s$ we use an Agglomerative Information Bottleneck (AIB) approach.

4.3.3 AIB compression

AIB [62] iteratively compresses the vocabulary W_s by merging the visual-words w_i and w_j which cause the smallest decrease in MI, $I(A, W_s)$. The algorithm can be summarized as follows:

- Initiate $\widehat{W}_s \equiv W_s$, i.e., by taking each video-word of W_s as a singleton cluster.
- Pair-wise distance computation: for every $\{w_i, w_j\} \in \widehat{W}_s$, $i < j$, the distance d_{ij} (which is a measure of MI) is computed:

$$d_{ij} = (p(w_i) + p(w_j)) \cdot JS_{\Pi}[p(a|w_i), p(a|w_j)] \tag{4.8}$$

where $JS_{\Pi}[p(a|w_i), p(a|w_j)]$ is the Jensen-Shannon divergence for a M class distribution, $p_i(x)$, each with a prior π_i, and is defined as:

$$JS_{\Pi}[p_1, p_2, \ldots, p_M] \equiv H[\sum_{i=1}^{M} \pi_i p_i(x)] - \sum_{i=1}^{M} \pi_i H[p_i(x)] \tag{4.9}$$

where $H[p(x)]$ is Shannon's entropy:

$$H[p(x)] = -\sum_{x} p(x) \log p(x) \tag{4.10}$$

- Merging: select the pair of video-words $\{w_\alpha, w_\beta\}$ for which the distance $d_{\alpha\beta}$ is minimum and merge them. Hence, we merge the video-words which result in the minimum MI loss by optimizing the global criterion in Equation 6.12.

AIB is a greedy algorithm in nature and optimizes the merging of only two word clusters at every step (local optimization). Hence, it optimizes the global criteria defined in Equation 6.12. We use the described vocabulary compression at each level of the pyramid per class, and obtain a final class-specific compact pyramid representation of video-words.

We use AIB for the vocabulary compression instead of Principal Component Analysis (PCA) based dimensionality reduction, since PCA is a linear model, whereas the relationship among the video words are highly non-linear in nature. Besides, PCA based dimensionality reduction will work on the first level cluster (k-means) of the bag-of-words model to reduce the final bag-of-words histogram dimensionality. Hence, it will not take inter and intra cluster similarities into account. Unlike PCA, the agglomerative information bottleneck (AIB) method presented in the article, is non-linear and it yields a set of compressed clusters from the first level clusters, such that the set of resulting compressed clusters maximally preserves the original information among them. Additionally, AIB based compression explores the mutual information present among video words and apply compression based on this information. Hence, in this case, AIB based compression is analytically more appropriate than PCA.

To empirically support our selection of AIB based compression, we have conducted experiments on the Weizmann dataset using PCA based dimensionality reduction. The obtained average accuracy is quite low (∼40% in the range of 30%–70% compression) compared to the recognition rate of AIB (∼99% in the same range of compression), which documents that AIB is a far better choice.

4.3.4 Action classification

After compression of the video-words at each pyramid level we compute a histograms of the video-words, using the extracted local motion features, and concatenate them to a final feature set for SVM learning. We design a class specific χ-square kernel-based SVM, $\text{SVM}_{a_i}(k, h^{a_i}_{W_{a_i}})$ [9], where a_i is the i^{th} action class A, k is the SVM kernel and $h^{a_i}_{W_{a_i}}$ is the histogram of action class a_i, computed using the class-specific video-words W_{a_i}. For a test set a_{Test} we detect its action class:

$$i^*_{a_{Test}} = argmax_j \text{SVM}_{a_j}(k, h^{a_{Test}}_{W_{a_j}}), \forall a_j \in A \tag{4.11}$$

We conduct experiments using different SVM kernels, and observe that the χ-square and intersection kernel are the best perfoming SVM kernels for all the datasets. Hence, we apply the χ-square kernel for all our experiments on human action recognition in section 7.5. Table 4.1 shows the average recognition accuracy for the Weizmann dataset using a number of different SVM kernels.

Table 4.1: Average recognition accuracy for the Weizmann dataset using different SVM kernels. We have used a Polynomial kernel of degree 3.

SVM Kernel	Recognition rate (%)
χ-square	**99.50**
Intersection	97.78
Radial basis function	87.77
Polynomial	78.67
Linear	58.89

4.4 Experimental results

4.4.1 Human action datasets

To test our proposed approach for action recognition we conduct a comprehensive set of experiments using a number of publicly available human action datasets (see Figure 4.1), which are categorized as follows.

Single actor benchmark

To conduct benchmark testing we choose the two most popular human action datasets: KTH [59] and Weizmann [19]. Both of these datasets contain single actors and clean backgrounds. The KTH dataset consists of 6 different actions: *walking*, *jogging*, *running*, *boxing*, *clapping* and *waving*. These actions are performed in 4 different but well-controlled environments by 25 different actors, resulting in a total of 600 action instances. The Weizmann dataset contains 90 videos separated into 10 actions performed by 9 persons. The actions are: *bend*, *jumping-jacks*, *jump*, *jump-in-place*, *run*, *gallop-sideways*, *skip*, *walk*, *one-hand-waving* and *two-hands-waving*.

Single actor with complex background

In this category we choose the CVC action dataset [1] and the CMU action dataset [27]. The CVC dataset consists of 5 actors performing 7 actions: *walking*, *jogging*, *running* (with horizontal and vertical two-way paths), *hand-waving*, *two-hands-waving*, *jump-in-place* and *bending*. The dataset is rated "semi-complex" and is interesting, since it has a textured background. The CMU dataset is composed of 48 video sequences of five action classes: *jumping-jacks*, *pick-up*, *push-button*, *one-hand-waving* and *two-hands-waving*. The test data contains 110 videos (events) which are down-scaled to 160×120 in resolution. This dataset has been recorded by a hand-held camera with moving people and vehicles in the background, and is known to be very challenging.

Movie and YouTube video clips

To evaluate our approach in different challenging stettings, we conduct experiments on movie and YouTube video clips. Concretely, we use the Hollywood 2 human actions and

scenes dataset [45] and the YouTube action dataset [39]. The Hollywood 2 dataset is composed of video clips extracted from 69 Hollywood movies, and contains 12 classes of human actions: *AnswerPhone*, *DriveCar*, *Eat*, *FightPerson*, *GetOutCar*, *HandShake*, *HugPerson*, *Kiss*, *Run*, *SitDown*, *SitUp* and *StandUp*. In total, there are 1707 action samples divided into a training set (823 sequences) and a test set (884 sequences), where train and test sequences are obtained from different movies. The dataset intends to provide a comprehensive benchmark for human action recognition in realistic and challenging settings. The YouTube dataset is a collection of 1168 complex and challenging YouTube videos of 11 human actions categories: *basketball shooting*, *volleyball spiking*, *trampoline jumping*, *soccer juggling*, *horseback riding*, *cycling*, *diving*, *swinging*, *golf swinging*, *tennis swinging* and *walking (with a dog)*. The dataset has the following properties: a mix of steady cameras and shaky cameras, cluttered background, low resolution, and variation in object scale, viewpoint and illumination. The first four actions are easily confused with jumping, the next two may have similar camera motion, and all the swing actions share some common motions. Some actions are also performed with objects such as a horse, bike or dog.

Multiple actors with complex background

We use two multiple actor datasets: the Microsoft research action dataset I (MSR I) [78] and the Multi-KTH dataset [67]. MSR I consists of 16 video sequences and a total of 63 actions: 14 *hand-clapping*, 24 *hand-waving* and 25 *boxing*, performed by 10 subjects. The sequences contain multiple types of action recorded in indoor and outdoor scenes with cluttered and moving backgrounds. Some sequences contain multiple actions performed by different people. Each video is of low resolution 320 × 240 with a frame rate of 15 frames per second, and their lengths are between 32 to 76 seconds. The Multi-KTH dataset is a more challenging version of the KTH dataset. It contains 5 (except *running*) of the 6 KTH-actions, which have been recorded by a hand-held camera, with multiple simultaneous actors, a significant amount of camera motion, scale changes and a more realistic cluttered background.

4.4.2 Automatic action annotation for Multi-KTH

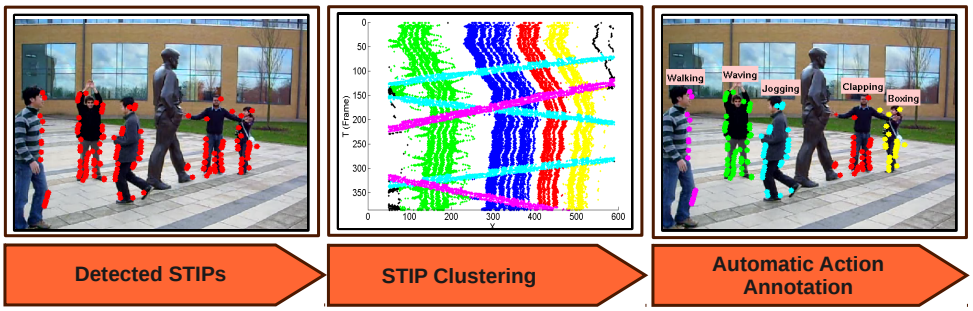


Figure 4.9: A schematic overview of the spatio-temporal clustering module and the associated data flow pipeline.

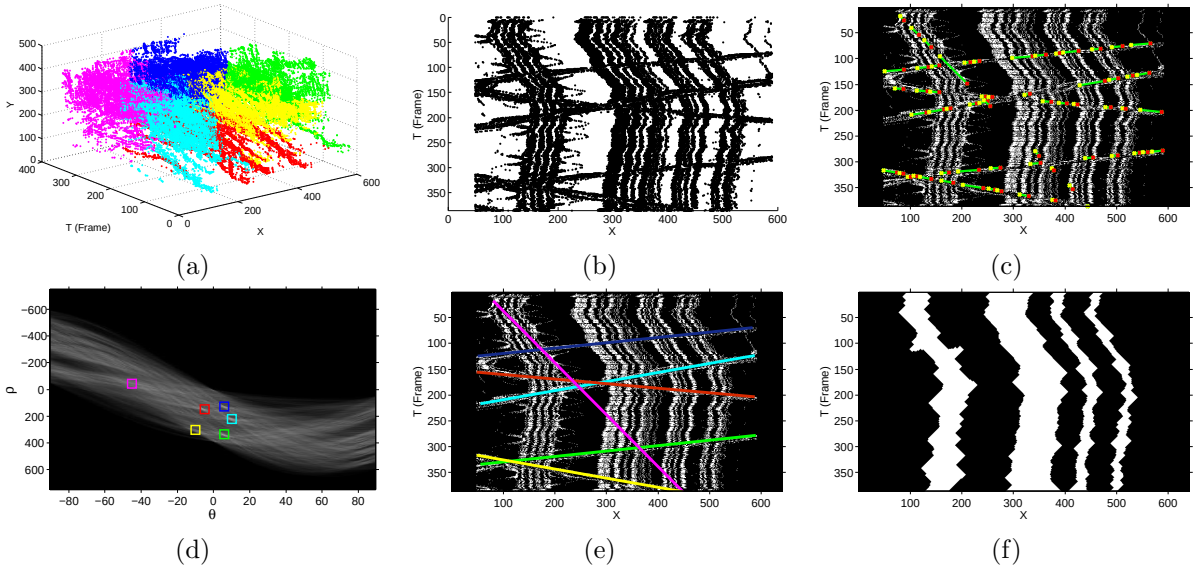


Figure 4.10: Plots of the detected STIPs for the Multi-KTH dataset, and detection of linear patterns in the XT-space. (a) k-means clustered STIPs in the $3D$ spatio-temporal XYT-space and (b) ungrouped STIPs in the $2D$ spatio-temporal XT-space; (c) line segments in XT-space caused by actions like walking, jogging or running; (d) candidates with high responses in the Hough space; (e) detected line segment using the Hough transfrom and (f) *blobs* obtained by morphological operations.

When multiple actors appear simultaneously in a scene, it is necessary to group the detected STIPs into actor-specific clusters. An excellent example is the Multi-KTH dataset, where five actors are present in the scene. Based on this dataset we introduce a spatio-temporal clustering technique for actor-specific STIP grouping and evaluate its performance in section 4.4.8. This spatio-temporal clustering is only a part of Multi-KTH dataset for automatic annotation.

Actor-specific STIP clustering

The actions present in the Multi-KTH dataset can be divided into two main groups: the actions with moving actors, like *walking* and *jogging*, and the actions with static actors, like *boxing*, *waving* and *clapping*. These two different nature of actions can be analyzed in the 2D spatio-temporal XT-space (see Figure 4.10.b). The actor-specific STIP clustering exploits the 2D spatio-temporal XT-space and consist of two main steps:

1. detection of lines in the XT-space and cluster STIPs accordingly,

2. after the first set of STIP clusters have been estimated, the associated STIPs are excluded and the resulting subset is clustered using morphological operations and a spatio-temporal distance measurement.

The surround suppression effect of our STIP detector, resulting in a low detection rate of unwanted background STIPs, facilitates STIP clustering in the XT-space. This will

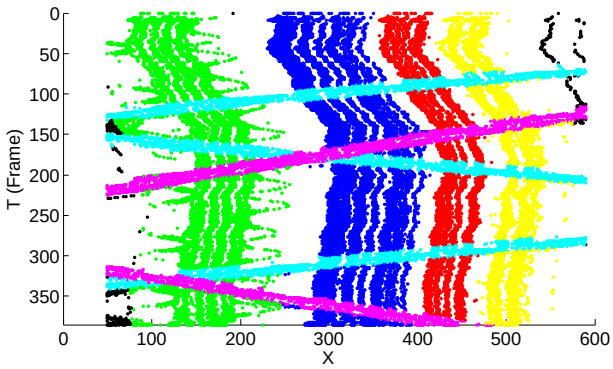
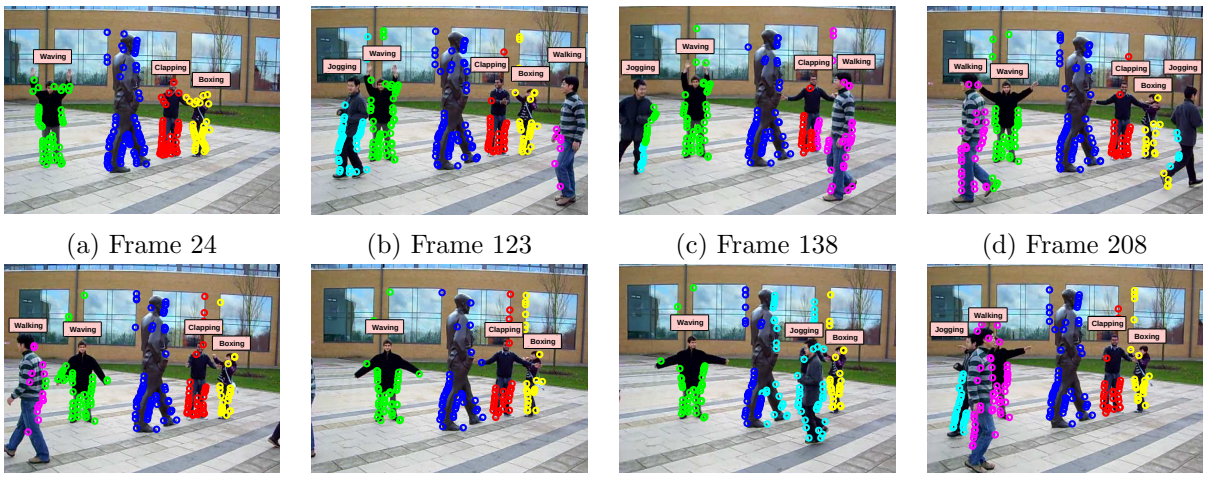


Figure 4.11: Actor-specific STIP clustering in the XT-space.



(a) Frame 24 (b) Frame 123 (c) Frame 138 (d) Frame 208

(e) Frame 217 (f) Frame 234 (g) Frame 296 (h) Frame 333

Figure 4.12: Automatic annotation of STIPs detected for multiple simultaneously actors for a number of frames from the Multi KTH dataset.

simply not be possible with a high number of background STIPs. Figure 4.9 illustrates the concept of the spatio-temporal clustering.

The spatio-temporal XT-space

A plot of the detected STIPs in 3D spatio-temporal XYT-space for the Multi-KTH sequence is shown in Figure 4.10.a. As can be seen, actor-specific clustering of the STIPs is non-trivial due to camera motion and occlusions. Hence, successful clustering cannot be accomplished by commonly used methods, e.g., k-means or Mean Shift clustering. Instead, we project the 3D spatio-temporal STIPs onto a 2D spatio-temporal XT-space, as shown in Figure 4.10.b, which reveals some interesting and useful patterns. The XT-space can be seen as the top-down view of the 3D spatio-temporal XYT-space (Figure 4.10.a), with the horizontal and vertical axes representing the X-position and the time T, respectively. Hence, the T-axis demonstrates the evolution of STIPs in time.

Table 4.2: STIP detection ratios (%): the number of STIPs detected on the actors with respect to the total number of detected STIPs, estimated for the MSR I and Multi-KTH datasets using our approach and state-of-the-art methods.

Method	MSR I	Multi-KTH
Our approach	**76.21**	**90.34**
Laptev el al. [34]	18.73	48.16
Dollár et al. [11]	21.36	16.03
Willems et al. [71]	24.02	20.24

Detection of lines in XT-space

Actions like *walking*, *jogging* or *running* create lines in the XT-space. Hence, we detect line segments in XT-space to cluster STIPs detected for the actors. This is valid, since actors with a certain target destination move in a linear pattern for those actions. Hough transform [13] is applied for the detection of these linear patterns (i.e., line segments) and the candidates with high response in the Hough Space are kept. Furthermore, a post candidate approval is applied based on the slope of the lines. Figure 4.10 shows this process and the intermediate results. As can been seen, the erroneously detected (magenta colored) line can be discarded according to its steep slope. Furthermore, Line segments for the crossing actors are detected but due to a high amount of camera motion, it is not possible to detect good candidates for the other actors performing upper body acations, like *boxing*, *clapping* and *waving*.

STIP clustering in XT-space

We use the detected lines to cluster the STIPs by applying a point-line distance measure $d(x,t)$, and threshold according to a maximum distance d_{max} for each line segment:

$$d(x,t) = \frac{|(\mathbf{p}-\mathbf{q_1}) \times (\mathbf{p}-\mathbf{q_2})|}{|\mathbf{q_2}-\mathbf{q_1}|} < d_{max} \tag{4.12}$$

where $\mathbf{p}$ is the current STIP under evaluation, and $\mathbf{q_1}$ and $\mathbf{q_2}$ are two points lying on a detected line. The maximum distance d_{max} is set according to the size of the actors appearing in the dataset. After clustering the first set of STIPs, we exclude them and use the remaining STIPs for further clustering. We merge the new subset of STIPs by morphological operations (see Figure 4.10.f) and use the resulting *blobs* to cluster the STIPs, by considering the spatio-temporal distance between a STIP and the contours. Figure 4.11 shows the resulting actor-specific STIP clustering in the XT-space, and in figure 4.12 the grouped STIPs are superimposed on a number of frames from the Multi-KTH dataset.

4.4.3 Evaluation of STIP detector

We evaluate our STIP detector by estimating a score for the number of detected STIPs for the actors in comparison to those detected in the background. Cao et al. [8] have

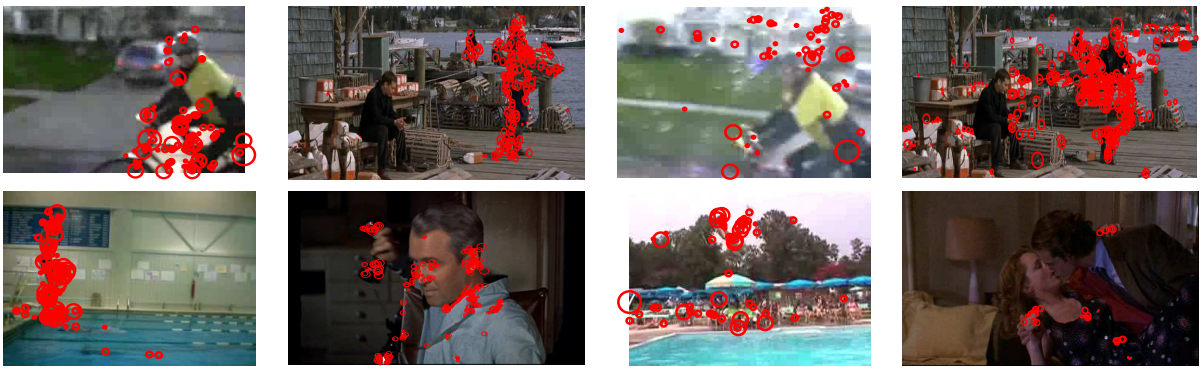
recently reported that of all the STIPs detected by Laptev's STIP detector [34], only 18.73% correspond to the three actions performed by the actors in the MSR I, while the rest of the STIPs (81.27%) belong to the background. Ground truth bounding boxes are used to determine if a STIP belongs to an action instance. We evaluate our STIP detector on MSR I in a similar way, and detect **76.21**% STIPs for the actors. We observe that our detector tends to detect more points in the background, when applied to the sequences of MSR I with several moving people in the background. Our STIP detector is designed to detect interest point for people, hence it will also consider moving people in the background as candidates. We also conduct this experiment for the Multi-KTH dataset by manually annotating ground truth bounding boxes, and find that **89.35**% STIPs belong to the actors (see Figure 4.4). This is consistent with the concept of our STIP detector, and documents the effectiveness of our incorporated surround suppression followed up by imposing local and temporal constraints. Table 4.2 shows STIP detection ratios of the state-of-the-art methods, and clearly documents the superior performance of our STIP detector.

The time complexity of our STIP computation highly depends on the size of the input video. For a video of size ($160 \times 120 \times 550$), the STIP computation, executed on a standard dual core Desktop PC (Intel(R) Core(TM)2 CPU 6400@2.13 GHz 6 GB RAM) using MATLAB R2010, takes approximately 10 min.

Figure 4.13 shows the perfomance of the STIP detector in complex scenarios. Despite of the camera movement, the STIP detector performs well (Figure 4.13(a) and (b)). However, in some cases, due to the combination of complex backgorund, low resolution and large background motion, the STIP detector loses focus and detects a larger number of background STIPs (Figure 4.13(c)) or an insufficient number of actor STIPs (Figure 4.13(d)).

4.4.4 Vocabulary building

The purpose of this experiment is to reveal the optimal initial vocabulary size and compression rate for our vocabulary building strategy. We divide each dataset into 50% training, 20% validation and 30% testing partitions. The final training of the SVMs uses both the training and validation sets. The recognition rates are computed by averaging over 50 random instances of these sets. We conduct experiments using a similar vocabulary size range as Liu et al. [39], with an initial vocabulary size of 50 video-words and incrementing it up to 400. We weight the initial vocabulary size according to the pyramid level using a weight factor 2^{-s}, where s is the pyramid level. The vocabulary size is weighted to avoid the empty/singleton cluster creation in finer levels of the pyramid. We reduce the dimensionality of the final feature vectors for the SVM classifiers by applying vocabulary compression at each pyramid level. To choose the optimal vocabulary size and compression rate, we vary the initial vocabulary size range [50–400] with an increment of 20, and for each of these vocabularies we vary the compression rate from 0% to 95% with an increment of 5%. Figure 4.14.a shows the resulting 3D plot of the recognition rate as a function of the initial vocabulary size and the compression rate, for the KTH dataset. The maximum recognition rate indicates the optimal vocabulary size and compression rate. We observe that the best result is obtained at a compression rate upto **65**%, and



(a) (b) (c) (d)

Figure 4.13: Performance of the STIP detector in sequences with complex scenarios. Successful STIP detection is shown for frames of the (a) YouTube and (b) Hollywood 2 dataset, respectively. Additionally, the failure frames of (c) YouTube and (d) Hollywood 2 are also shown. In (a) and (b) our STIP detector successfully handles camera motion and the STIPs are detected only in the motion of interest. On the contrary, in the frames of (c) and (d), due to high background motion and difference in scene resolution the STIP detector loses the focus on the motion of the human actors.

the performance starts to degrade rapidly above **80**%. In Figure 4.14.b the recognition rate, as a function of the initial vocabulary size for the three other single actor datasets: CMU, CVC and Weizmann, is shown. We obtain approximately 100% recognition rate in the initial vocabulary size range [230–300] for the Weizmann, CMU and CVC datasets, which is similar to the the middle peak in Figure 4.14.a for KTH.

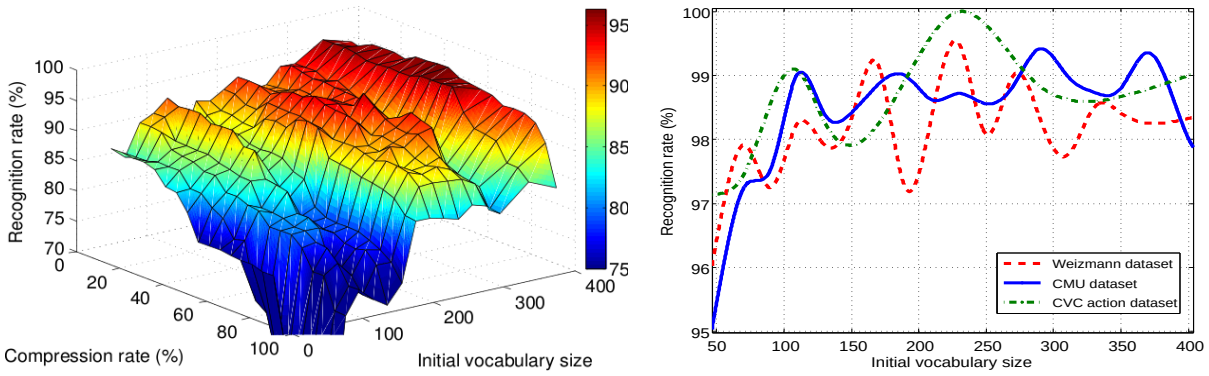
4.4.5 Benchmark testing

We use the KTH and Weizmann datasets for benchmark testing, and achieve an accuracy of **96.35**% for KTH and **99.50**% for Weizmann. Table 4.3 shows a comparison of the recognition rates of our approach and several other state-of-the-art methods for these two datasets. It should be noted that we achieve state-of-the-art recognition rate for KTH. We obtained this recognition with an initial vocabulary size of 350 and a 60% compression rate. The main reasons for this improvement are the selective STIP detection and the spatial pyramids, which capture the local characteristics of actions, and thereby reduce interclass confusion. The accuracy for Weizmann is approximately 100%, which is comparable to the state-of-the-art. Lin et al. [37] report a clear 100% recognition rate for Weizmann. However, this work applies a template matching technique, using holistic features extracted from global boundary box-based interest regions. Furthermore, it requires background subtraction and target tracking. In contrast, our approach uses local features and does not require any preprocessing. Since, Weizmann is a simple datasets without any further challenges, it favors global and holistic methods. In contrast, our approach is applicable for all types of scenes, including very challenging scenes of high complexity, which we will validate in the following.

Table 4.3: State-of-the-art recognition accuracies (%) for the KTH, Weizmann and YouTube datasets. *Liu et al. [41] test on 8 out of the 11 YouTube actions.

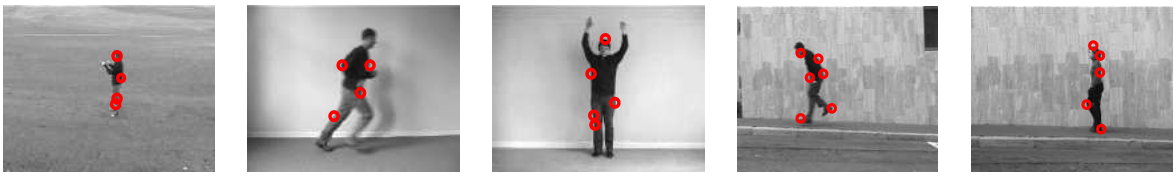
Method	KTH	Weiz.	YouTube
Our approach	**96.35**	99.50	**86.98**
Lui et al. [43]	96.00	-	-
Yu et al. [77]	95.67	-	-
Kim et al. [28]	95.33	-	-
Wu et al. [73]	95.10	98.90	-
Cao et al. [8]	95.02	-	-
Kaâniche et al. [26]	94.67	-	-
Kovashka et al. [31]	94.53	-	-
Gilbert et al. [17]	94.50	-	-
Sadek et al. [55]	94.30	-	-
Liu & Shah [40]	94.16	-	-
Sun et al. [64]	94.00	97.80	-
Saghafi et al. [56]	93.94	-	-
Shao et al. [61]	93.89	-	-
Liu et al. [39]	93.80	-	71.20
Uemura et al. [67]	93.70	-	-
Lin et al. [37]	93.43	**100.00**	-
Yuan et al. [78]	93.30	-	-
Liu et al. [41]	92.30	-	76.10*
Yao et al. [76]	93.00	92.20	-
Schindler et al. [57]	92.70	**100.00**	-
Laptev et al. [33]	91.80	-	-
Jhuang et al. [24]	91.70	98.80	-
kläser et al. [29]	91.40	84.30	-
Yang et al. [75]	87.30	99.40	-
Wong et al. [72]	86.62	-	-
Willems et al. [71]	84.26	-	-
Niebles et al. [50]	81.50	-	-
Dollár et al. [11]	81.17	-	-
Schüldt et al. [59]	71.72	-	-
Gorelick et al. [19]	-	99.64	-
Thurau et al. [65]	-	94.40	-
Ali et al. [2]	-	92.60	-
bregonzio et al. [6]	-	-	64.00

We analyze the error-frames of the 0.50% videos of the Weizmann dataset, which are miss-classified. Similarly, we analyze the miss-classified frames from the confusion matrix for KTH. Figure 4.15 shows some example error-frames. Due to low resolution only a limited number of STIPs are detected for the important body parts (arms and legs), which are taking major part in actions like *boxing* and *running*. In these few cases this results in miss-classification.



(a) (b)

Figure 4.14: Revealing the influence of the vocabulary size and compression on the average action recognition rates. (a) A 3D Plot of the recognition rate, as a function of the initial vocabulary size and the compression rate, for the KTH dataset. (b) Recognition rates, as a function of the initial vocabulary size, for the three single actor datasets: CMU, CVC and Weizmann. The compression rate is fixed to 65%, i.e., 35% of the initial vocabulary size is used.



(a) KTH *boxing* (b) KTH *running* (c) KTH *waving* (d) Weiz. *skip* (e) Weiz. *walking*

Figure 4.15: Error-frames of the videos that are miss-classified for the KTH and Weizmann datasets. The first three frames depict miss-classified *boxing*, *running* and *waving* actions from the KTH dataset, respectively. The last two error-frames are *skip* and *walking* actions from the Weizmann dataset. These frames show cases which result in miss-classification. Due to low resolution only a limited number of STIPs are detected for the important body parts (arms and legs), which are taking major part in these actions.

4.4.6 Evaluation on complex scene

The main objective of this evaluation is to test the capability of our method to handle background clutter. For this purpose we choose the CMU action dataset and the CVC Action dataset with textured background. Despite the presence of strong background texture and clutter, we achieve a **100.0**% accuracy rate for CVC and 99.42% for CMU (see Table 4.5). The high performance for both of these dataset is consistence with the theoretical foundation of our proposed STIP detector. The detector's selective behavior, achieved by incorporating surround suppression and imposing local and temporal constraints, results in robustness to background texture and clutter.

Table 4.4: The average precision (%) and mean average precision (MAP) for the actions of Hollywood 2, using our approach in comparison to the state-of-the-art.

Action	Marszalek [45]	Han [21]	Wang [69]	Gilbert [17]	Ullah [68]	**Our**
AnswerPhone	13.10	15.57	-	40.20	26.30	**41.60**
DriveCar	81.00	87.01	-	75.00	86.50	**88.49**
Eat	30.60	50.93	-	51.50	**59.20**	56.50
FightPerson	62.50	73.08	-	77.10	76.20	**78.20**
GetOutCar	8.60	27.19	-	45.60	45.70	**47.37**
HandShake	19.10	17.17	-	28.90	49.70	**52.50**
HugPerson	17.00	27.22	-	49.40	45.40	**50.30**
Kiss	57.60	42.91	-	56.60	**59.00**	57.35
Run	55.50	66.94	-	47.50	72.00	**76.73**
SitDown	30.00	41.61	-	62.00	62.40	**62.50**
SitUp	17.80	7.19	-	26.80	27.50	**30.00**
StandUp	33.50	48.61	-	50.70	58.80	**60.00**
MAP	35.50	42.12	47.70	50.90	55.70	**58.46**

Table 4.5: Recognition accuracies (%) for cross-data evaluation trained on KTH and tested on other datasets: Weizmann, CVC, CMU, MSR I and Multi-KTH. The first row presents results when training and testing on the same dataset for Weizmann, CVC and CMU.

Method	Weizmann	CVC	CMU	MSR I	Multi-KTH
Our approach (÷ cross-data)	**99.50**	**100.00**	**99.42**	-	-
Our approach	**100.0**	**96.95**	**91.94**	**84.77**	**98.40**
Yuan et al. [78]	-	-	70.00	-	-
Cao et al. [8]	-	-	-	60.00	-
Gilbert et al. [18]	-	-	-	-	75.20
Gilbert et al. [17]	-	-	-	-	68.80
Uemura et al. [67]	-	-	-	-	65.40

4.4.7 Action recognition in movie and YouTube video clips

Next, we conduct experiments on movie and YouTube video clips, using the YouTube and Hollywood 2 action datasets. We achieve 99.13% recognition rate for the YouTube actions. Table 4.3 shows the comparison with other state-of-the-art method for this dataset. Our approach is far superior compared to the other reported methods, due to our STIP detector's capability to handle complex and challenging scenes with camera motion, cluttered background, and variation in scale, viewpoint and illumination.

For the Hollywood 2 dataset, the performance is evaluated as suggested in [45], i.e., by computing the average precision (AP) for each of the action classes and reporting the mean AP over all classes (MAP). Table 4.4 shows the AP for the actions in comparison to other state-of-the-art methods. The Hollywood 2 dataset contains very complex scenes from movies with no ground truth information available, and moreover the different instances of an action are sometimes viewed from different camera angles.

Notes: "*Answerphone* and *Handshake* are quite small, and therefore need a very complex set of compound features in order to classify the action over the background noise. In contrast, *FightPerson* and *DriveCar* use more global contextual features and therefore they work with lower level features."

4.4.8 Cross-data experiments

We perform exhaustive cross-data evaluation to test our proposed method in more realistic scenarios and use the KTH and Weizmann datasets for training data. We observe that the Weizmann dataset is not appropriate for training, and results in a poor 40% and 45% recognition rate for CVC and CMU, respectively. This is due to inadequate training data since Weizmann contains a very limited number of action instances per category compared to KTH. Table 4.5 shows the accuracy rates obtained using KTH as training. These cross-data results validate that our approach is applicable for more practical scenarios, where training and test data are coming from different sources.

The KTH dataset has only one common action, *two-hands-waving*, with the CMU action dataset. We use the KTH *running* sequence as negative data and obtain a **91**.**94**% recognition rate. It is noticeable, that the accuracy is actually higher for Weizmann (**100**%) and CMU (**99**.**42**%), than when training and testing on the same dataset, due to the sufficient action instances for training. Additionally, for CMU we only recognize one action, *two-hands-waving*, compared to five actions when both training and testing on CMU. On the contrary, the accuracy decreases by 3% for CVC, due to its lower inter-dataset correlation with KTH. For the Multi-KTH dataset we manually annotate the action labels as ground truth, using bounding boxes, and obtain **98**.**40**% accuracy. We perform another test using our automatic action annotation described in section 4.4.2, and obtain a 94.20% recognition rate, which is comparable to the results of the manual annotation. For the MSR I dataset we achieve **84**.**77**% accuracy. The difficult part of MSR I is that some sequences contain moving people in the background depicted by the bounding box of the agent performing the action, which result in unwanted STIP in the background, and thereby a lower recognition rate compared to the other datasets. In conclusion, these results outperform the state-of-the-art significantly (see Table 4.5) and hereby validate the robustness of our method in more realistic action recognition scenarios. Although these datasets are very complex and contain several practical challenges: cluttered and moving backgrounds (including people and vehicles), camera motion and multiple actors, our approach performs robustly.

4.5 Conclusion

In this paper we have presented a novel approach for human action recognition in complex scenes. Our approach is based on selective STIPs which are detected by suppressing background SIPs and imposing local and temporal constraints, resulting in more robust STIPs for actors and less unwanted background STIPs. We apply a BoV model of local N-jet descriptors extracted at the detected STIPs and introduce a novel vocabulary building strategy by combining a spatial pyramid and vocabulary compression. Action

class-specific SVM classifiers are trained to finally identify human actions.

The strong aspect of our proposed STIP detection method is, it can detect dense STIPs at the motion region without affected by the complex background. This is an important property to detect actions in complex scenarios. Regarding the weak aspect, our method suffers in the presence of other motion (presence of multiple actors) together with the region of action. In this scenario we detect several STIPs from different motion region results in poor classification.

In the current system, we use greedy approach for vocabulary compression. Sometimes, the time complexity is higher with this approach. A non-greedy method for vocabulary compression might be an interesting inclusion for the future work. Our automatic action annotation using STIP clustering works well for the multi-KTH dataset, yet it is not generalized for other multi-actor action datasets. The automatic action annotation for multi-actor datasets is a very difficult and challenging task. We could include more complex shape matching algorithm along with a human model in the XT-space to minimize the overlap in the STIP clusters of the moving and non-moving actors.

We have reported superior action recognition results in comparison to the state-of-the-art, when testing on benchmark datasets of simple scenes (96.35% accuracy for KTH and 99.50% for Weizmann), and similar performance for complex scenes (CVC and CMU). Additionally, we have shown state-of-the-art performance and proven the applicability of our approach for action recognition in movie and YouTube video clips by significantly outperforming other methods evaluated on the YouTube action dataset, and showing the highest mean average precision for the Hollywood 2 dataset. A comprehensive cross-data evaluation has been performed by separating the training (KTH) and test datasets (CVC, CMU, MSR I and Multi-KTH). To our best knowledge we are the first to report exhaustive cross-data evaluation. Compared to state-of-the-art we have reported superior results by raising the recognition rates from approximately 60–75% to 85–100%.

Acknowledgments

This work has been supported by the Spanish Research Programs Consolider-Ingenio 2010:MIPRCV (CSD200700018); Avanza I+D ViCoMo (TSI-020400-2009-133); the Spanish Project TIN2009-14501-C02-02; and the Danish National Research Councils – FTP under the research project "Big Brother *is* watching you!".

References

[1] the CVC dataset is available at http://iselab.cvc.uab.es/files/Tools/Cvc-ActionDataSet/index.htm.

[2] S. Ali, A. Basharat, and M. Shah. Chaotic invariants for human action recognition. In *ICCV*, 2007.

[3] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006.

[4] P. Beaudet. Rotationally invariant image operators. In *ICPR*, 1978.

[5] O. Boiman and M. Irani. Detecting irregularities in images and in video. In *ICCV*, 2005.

[6] M. Bregonzio, J. Li, S. Gong, and T. Xiang. Discriminative topics modelling for action feature selection and recognition. In *BMVC*, 2010.

[7] L. Bretzner and T. Lindeberg. Feature tracking with automatic selection of spatial scales. *CVIU*, 71(3):385–392, 1998.

[8] L. Cao, Z. Liu, and T.S. Huang. Cross-dataset action detection. In *CVPR*, 2010.

[9] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[10] Chuong B. Do and Andrew Y. Ng. Transfer learning for text classification. *Journal of Advances in Neural Information Processing Systems*, 18:299–306, 2006.

[11] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.

[12] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *ICCV*, 2009.

[13] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[14] Jialue Fan, Ying Wu, and Shengyang Dai. Discriminative spatial attention for robust tracking. In *ECCV*, pages 480–493, 2010.

[15] A. Galata, N. Johnson, and D. Hogg. Learning variable-length markov models of behavior. *CVIU*, 81(3):398–413, 2001.

[16] Dashan Gao, Sunhyoung Han, and Nuno Vasconcelos. Discriminant saliency, the detection of suspicious coincidences, and applications to visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:989–1005, 2009.

[17] A. Gilbert, J. Illingworth, and R. Bowden. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *ICCV*, 2009.

[18] A. Gilbert, J. Illingworth, and R. Bowden. Action recognition using mined hierarchical compound features. *PAMI*, 99(PrePrints), 2010.

[19] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *PAMI*, 29(12):2247–2253, 2007. the Weizmann dataset is available at http://www.wisdom.weizmann.ac.il/∼vision/ SpaceTime Actions.html.

[20] Cosmin Grigorescu, Nicolai Petkov, and Michel A. Westenberg. Contour and boundary detection improved by surround suppression of texture edges. *IVC*, 22(8):609–622, 2004.

[21] D. Han, L. Bo, and C. Sminchisescu. Selection and context for action recognition. In *ICCV*, 2009.

[22] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.

[23] N. Ikizler and D. Forsyth. Searching video for complex activities with finite state models. In *CVPR*, 2007.

[24] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007.

[25] Z. Jiang, Z. Lin, and L.S. Davis. A tree-based approach to integrated action localization, recognition and segmentation. In *ECCV Workshops*, 2010.

[26] M. B. Kaâniche and F. Brémond. Gesture recognition by learning local motion signatures. In *CVPR*, 2010.

[27] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *ICCV*, 2007. the CMU dataset instructions are available at http://www.cs.cmu.edu/∼yke/video/#Dataset.

[28] T.K. Kim, S.F. Wong, and R. Cipolla. Tensor canonical correlation analysis for action classification. In *CVPR*, 2007.

[29] A. Kläser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008.

[30] J.J. Koenderink and A.J. Van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[31] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *CVPR*, 2010.

[32] I. Laptev. On space-time interest points. *IJCV*, 64(2/3):107–123, 2005.

[33] I. Laptev, B. Caputo, C. Schüldt, and T. Lindeberg. Local velocity-adapted motion events for spatio-temporal recognition. *IJCV*, 108(3):207–229, 2007.

[34] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003.

[35] I. Laptev and T. Lindeberg. Local descriptors for spatio-temporal recognition. In *First International Workshop on Spatial Coherence for Visual Motion Analysis*, 2004.

[36] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[37] Z. Lin, Z. Jiang, and L.S. Davis. Recognizing actions by shape-motion prototype trees. In *ICCV*, 2009.

[38] T. Lindeberg. Feature detection with automatic scale selection. *IJCV*, 30(2):79–116, 1998.

[39] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *CVPR*, 2009. the YouTube dataset is available at http://www.cs.ucf.edu/∼liujg/YouTube_Action_dataset.html.

[40] J. Liu and M. Shah. Learning human actions via information maximization. In *CVPR*, 2008.

[41] J. Liu, Y. Yang, and M. Shah. Learning semantic visual vocabularies using diffusion distance. In *CVPR*, 2009.

[42] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[43] Y. M. Lui, J. R. Beveridge, and M. Kirby. Action classification on product manifolds. In *CVPR*, 2010.

[44] Vijay Mahadevan and Nuno Vasconcelos. Spatiotemporal saliency in dynamic scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:171–177, 2010.

[45] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009. the Hollywood 2 dataset is available at http://www.irisa.fr/vista/ actions/hollywood2.

[46] Pyry Matikainen, Martial Hebert, and Rahul Sukthankar. Representing pairwise spatial and temporal relations for action recognition. In *ECCV*, pages 508–521, 2010.

[47] T.B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 104(2-3):90–126, 2006.

[48] F. Nater, H. Grabner, and L. Van Gool. Exploiting simple hierarchies for unsupervised human behavior analysis. In *CVPR*, 2010.

[49] N.T. Nguyen, D.Q. Phung, S. Venkatesh, and H. Bui. Learning and detecting activities from movement trajectories using the hierarchical hidden markov model. In *CVPR*, 2005.

[50] J.C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 79(3):299–318, 2008.

[51] A. Oikonomopoulos, I. Patras, and M. Pantic. Spatiotemporal salient points for visual recognition of human actions. *SMC-B*, 36(3):710–719, 2006.

[52] Ronald Poppe. A survey on vision-based human action recognition. *IVC*, 28(6):976–990, 2010.

[53] K. Prabhakar, S. Oh, P. Wang, G.D. Abowd, and J.M. Rehg. Temporal causality for the analysis of visual events. In *CVPR*, 2010.

[54] M.D. Rodriguez, J. Ahmed, and M. Shah. Action mach: A spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

[55] S. Sadek, A. Al-Hamadi, B. Michaelis, and U. Sayed. Toward robust action retrieval in video. In *BMVC*, 2010.

[56] B. Saghafi, E. Farahzadeh, D. Rajan, and A. Sluzek. Embedding visual words into concept space for action and scene recognition. In *BMVC*, 2010.

[57] K. Schindler and L. van Gool. Action snippets: How many frames does human action recognition require. In *CVPR*, 2008.

[58] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *IJCV*, 37(2):151–172, 2000.

[59] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, 2004. the KTH dataset is available at http://www.nada.kth.se/cvap/actions.

[60] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ACM International Conference on Multimedia*, 2007.

[61] L. Shao and R. Gao. A wavelet based local descriptor for human action recognition. In *BMVC*, 2010.

[62] N. Slonim and N. Tishby. Agglomerative information bottleneck. In *NIPS*, 1999.

[63] E. Soria, J. Martin, R. Magdalena, M. Martinez, and A. Serrano. *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, chapter 11, pages 242–264. IGI Global, 2009.

[64] X. Sun, M. Chen, and A. Hauptmann. Action recognition via local descriptors and holistic features. In *CVPR*, 2009.

[65] C. Thurau and V. Hlavac. Pose primitive based human action recognition in videos or still images. In *CVPR*, 2008.

[66] Panu Turcot and David G. Lowe. Better matching with fewer features: The selection of useful features in large database recognition problems. In *ICCV Workshops*, 2009.

[67] H. Uemura, S. Ishikawa, and K. Mikolajczyk. Feature tracking and motion compensation for action recognition. In *BMVC*, 2008. the Multi-KTH dataset is available at http://www.openvisor.org/ video_details.asp?idvideo=303.

[68] M.M. Ullah, S.N. Parizi, and I. Laptev. Improving bag-of-features action recognition with non-local cues. In *BMVC*, 2010.

[69] H. Wang, M.M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[70] Daniel Weinland, Rémi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *INRIA Report*, RR-7212:54–111, 2010.

[71] G. Willems, T. Tuytelaars, and L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, 2008.

[72] S.F. Wong and R. Cipolla. Extracting spatiotemporal interest points using global information. In *ICCV*, 2007.

[73] X. Wu, Wi Liang, and Y. Jia. Incremental discriminative-analysis of canonical correlations for action recognition. In *ICCV*, 2009.

[74] M. Yang, F. Lv, W. Xu, K. Yu, and Y. Gong. Human action detection by boosting efficient motion features. In *ICCV*, 2009.

[75] W. Yang, Y. Wang, and G. Mori. Recognizing human actions from still images with latent poses. In *CVPR*, 2010.

[76] A. Yao, J. Gall, and L. Van Gool. A hough transform-based voting framework for action recognition. In *CVPR*, 2010.

[77] T. Yu, T. Kim, and R. Cipolla. Real-time action recognition by spatiotemporal semantic and structural forests. In *BMVC*, 2010.

[78] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. In *CVPR*, 2009. the MSR dataset is available at http://research.microsoft.com/en-us/um/people/zliu/ActionRecoRsrc.

[79] L. Zelnik-Manor and M. Irani. Statistical analysis of dynamic actions. *PAMI*, 28(9):1530–1535, 2006.

Chapter 5

A Survey on Multi-View Human Action Recognition

This chapter consists of the paper "Human 3D Body Modeling, Pose Estimation and Activity Recognition from Multi-View Videos: Comparative Explorations of Recent Developments" [A]. The paper presents a review and comparative study of recent research on multi-view human 3D body modeling, pose estimation and activity recognition. The work has been conducted in collaboration with Prof. Mohan Trivedi and Cuong Tran from The Computer Vision and Robotic Research Laboratory (CVRR), University of California, San Diego (UCSD), who have high expertice in 3D body modelling and pose estimation. Reference [B] describes intermediate work resulting in the final outcome in [A].

References

A. M.B. Holte, C. Tran, M.M. Trivedi and T.B. Moeslund. Human 3D Body Modeling, Pose Estimation and Activity Recognition from Multi-View Videos: Comparative Explorations of Recent Developments. Submitted to *Journal of Selected Topics in Signal Processing, IEEE Signal Processing Society*, 2011.

B. M.B. Holte, C. Tran, M.M. Trivedi and T.B. Moeslund. Human Action Recognition using Multiple Views: A Comparative Perspective on Recent Developments. In *ACM Multimedia Joint Workshop on Human Gesture and Behavior Understanding, Association for Computing Machinery, Scottsdale, Arizona, USA*, December 2011.

Human 3D Body Modeling, Pose Estimation and Activity Recognition from Multi-View Videos: Comparative Explorations of Recent Developments

M.B. Holte, C. Tran, M.M. Trivedi and T.B. Moeslund

Abstract

This paper presents a review and comparative study of recent multi-view approaches for human 3D body modeling, pose estimation and activity recognition. We discuss the application domain of human body modeling, pose estimation and activity recognition and the associated requirements, covering: advanced Human-Computer Interaction (HCI), assisted living, gesture-based interactive games, intelligent driver assistance systems, movies, 3D TV and animation, physical therapy, autonomous mental development, smart environments, sport motion analysis, video surveillance and Video annotation. Next, we review and categorize recent approaches which have been proposed to comply with these requirements. We report a comparison of the most promising methods for multi-view human action recognition using two publicly available datasets: the INRIA Xmas Motion Acquisition Sequences (IXMAS) Multi-View Human Action Dataset and the i3DPost Multi-View Human Action and Interaction Dataset. To compare the proposed methods, we give a qualitative assessment of methods which cannot be compared quantitatively, and analyze some prominent 3D pose estimation techniques for application, where not only the performed action needs to be identified but a more detailed description of the body pose and joint configuration. Finally, we discuss some of the shortcomings of multi-view camera setups and outline our thoughts on future directions of 3D body pose estimation and human action recognition.

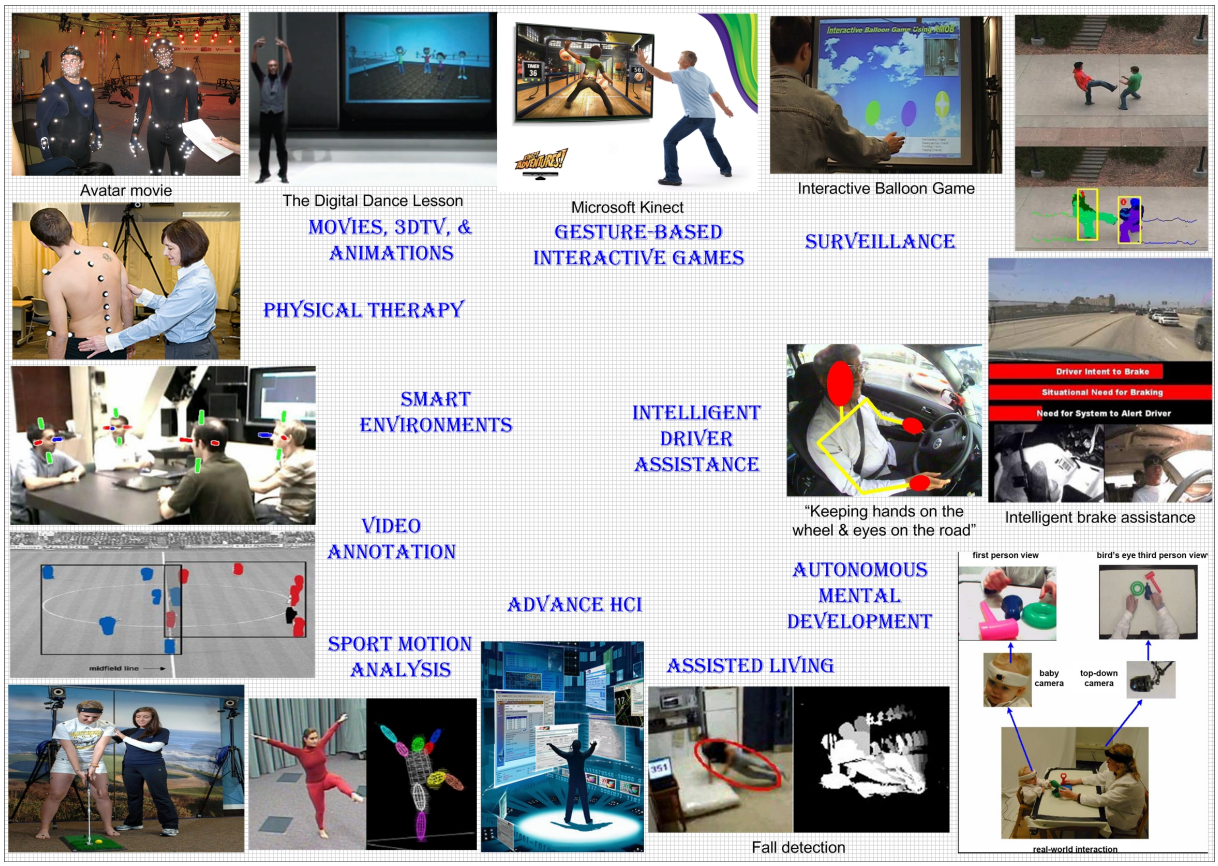


Figure 5.1: The application domain of human body modeling, pose estimation and activity recognition.

5.1 Introduction

"Looking at People" is a promising field within computer vision with many applications. Most rely on pose estimation (which again rely on human body modeling) and recognition. It is therefore interesting to get an overview of recent progress in these fields including how the different methods compare. In recent years a wide range of applications using 3D human body modeling, pose estimation and activity recognition has been introduced. Among those, several key applications are illustrated in Fig. 5.1 including:

- Advanced Human-Computer Interaction (HCI): Beyond traditional medium like computer mouse and keyboard, it is desirable to develop better, more natural interfaces between intelligent systems and human in which understanding visual human gesture is an important channel. A few examples are using hand movement to control the presentation slides [48] or recognizing manufacturing steps to help workers to learn and improve their skills [69].

- Assisted living: Pose estimation and activity recognition can also be applied in assisting handicapped people, elderly people, as well as normal people. For example a system to detect when a person falls [72] or a robot controlled by blinking [4].

- Gesture-based interactive games: In which the player use non-intrusive body move-

ment to interact with the games. For example an Interactive Balloon Game [86] or the well-known Microsoft Kinect Xbox [73].

- Intelligent driver assistance systems: Looking at driver is a key part required in a holistic approach for intelligent driver assistance systems [89]. Examples of driver assistance systems using posture and behavior analysis are: Monitoring driver awareness based on head pose tracking [61], combining driver head pose and hands tracking for distraction alert [85], modeling driver foot behavior to mitigate pedal misapplications [82], developing smart airbag system based on sitting posture analysis [90], or predicting driver turn intent [14].

- Movies, 3D TV and animation: Human motion capture is also applied extensively in movies, 3D TV and animation. For example in the Avatar movie, in a digital dance lesson [26] or for recording and representation of data for 3D TV [3].

- Physical therapy: Modern biomechanics and physical therapy applications require the accurate capture of normal and pathological human movement without the artifacts of intrusive marker-based motion capture systems. Therefore marker-less posture estimation and gesture analysis approaches were also developed to be applied in this area [70, 58]

- Autonomous mental development: Study the development of human mental capabilities by observing its real-time interactions with the environment using its own sensors and effectors, e.g. study the cognitive development and learning process of young children [11]. Instead of manually observing the data for analysis, such studies can utilize the recent advances in pose estimation and activity analysis to automate the process and enable analysis in a larger scale.

- Smart environments: In which humans and environment collaborate. Smart environments need to extract and maintain an awareness of a wide range of events and human activities occurring in these spaces [91]. For example, monitoring the focus of attention and interaction of participants in a meeting room [59, 95].

- Sport motion analysis: Several sports like golf, ballet, or skating require accurate body posture and movement therefore posture estimation and gesture analysis could be applied to this area for analyzing performance and training.

- Video surveillance: Video surveillance is used in many places such as critical infrastructure, public transportation, office buildings, parking lots, and homes. However manually monitoring these cameras is becoming a hazard. Therefore approaches for automatic video surveillance including outdoor human activity analysis, e.g. [63, 93] will be needed.

- Video annotation: With the development of hardware technology, a very large amount of video data can be easily saved. Among those, there are lots of human related videos such as surveillance videos, sport videos, or movies. Instead of manually scanning through those large video database to get the needed information, human motion analysis can be used to annotate those video, e.g. approaches to annotate video of a soccer game [6] or in more general for outdoor sports broadcasts [44].

Many approaches have been proposed to comply with the requirements of these applications, and based on different kinds of sensor systems for data acquisition: marker-based systems, laser-range scanners [100], structured light [24], Time-of-Flight (ToF) sensors [46, 80], the Microsoft Kinect sensor [73] and multi-camera systems [27]. Table 5.1 gives an overview of the application domain of human body modeling and motion analysis and the associated requirements. As can be seen, the requirements vary significantly depending on the desired application. This results in the need of approaches, which e.g. can operate on different abstraction levels, in uncontrolled environments, with high precision, in critical real-time and for large database search.

A number of surveys has been published during the last decade reviewing approaches for human motion capture, body modeling, pose estimation and activity recognition in more general [39, 56, 57, 67, 68, 99, 100]. This paper differs from these, in the sense that it focus exclusively on recent work on multi-view human body modeling, pose estimation and action recognition, both based on 2D multi-view data and reconstructed 3D data, acquired with standard cameras. Multi-view camera systems have the advantage that they enable full 3D reconstruction of the human body, and to some extent handles self-occlusion. In contrast single 3D imaging devices, like ToF sensors and Kinect, will only acquire 3D surface structure visible from that single viewpoint. We give a more detailed description and comparison of some prominent and diverse 3D pose estimation techniques, which represent the contributions to this field well. Additionally, we present a quantitative comparison of several promising multi-view human action recognition approaches using two publicly available datasets: the INRIA Xmas Motion Acquisition Sequences (IXMAS) Multi-View Human Action Dataset [96] and the i3DPost Multi-View Human Action and Interaction Dataset [27].

5.1.1 Human Body Modeling and Pose Estimation

Vision-based pose estimation and tracking of articulated human body is the problem of estimating kinematic parameters of the body model (such as joints position and joints angle) from a static image or a video sequence. Typically, the shape and dimension of body parts are assumed fixed and the interdependence between body parts are only the kinematic constraints at body joints. Related research studies in this area include body pose estimation, hand pose estimation and head pose estimation. Among those, the most extensive subfield is body pose estimation, which refers to the articulated body model normally with torso, head, and 4 limbs but without details of hand, foot, or facial variation. Several important applications explicitly required detailed 3D posture information including movies and 3D animation, sport motion analysis, physical therapy, as well as some application in advanced HCI or smart environments (e.g. robot controls or applications using pointing gesture). Moreover, the output 3D pose information is also a rich and view-invariant representation for action recognition [68]. Developing an efficient and robust body pose estimation system however is a challenging task. One major reason is the very high dimensionality of the pose configuration space, e.g. in [13], 19 DoF (Degree of Freedom) are used for the body model and 27 DoF are used for the hand model. As concluded in [75], although human tracking is considered mostly solved in constrained situations, i.e. has large number of calibrated camera (> 10), people wear tight clothes,

Table 5.1: Different applications and their requirements to 3D human body modeling, pose estimation and activity recognition approaches.

Application	Abstraction level	Data acquisition	Precision	Time critical	Human body model
Advanced HCI	PE/HAR,	Mostly controlled	Medium-high	Real-time	Mostly model-based
Gesture-based interactive games	PE/HAR	Mostly controlled	Medium-high	Real-time	Mostly model-based
Movies and 3D animation	PE	Controlled and uncontrolled	High	No	Model-based
Smart environments	PE/HAR	Controlled and uncontrolled	Medium	Real-time	Model-based and model-free
Video surveillance	HAR	Mostly uncontrolled	Low-medium	Real-time unless it is video search	Mostly model-free
Intelligent driver assistance	PE/HAR	Controlled and uncontrolled	High	Critical real-time	Model-based and model-free
Video annotation	HAR/PE	Uncontrolled	Low-medium	No but low computation time is desired for large scale databases	Mostly model-free
Sport motion analysis	PE	Controlled	High	No	Model-based
Physical therapy	PE	Controlled	High	No	Model-based
Assisted living	PE/HAR	Controlled and uncontrolled	Medium	Real-time	Model-based and model-free

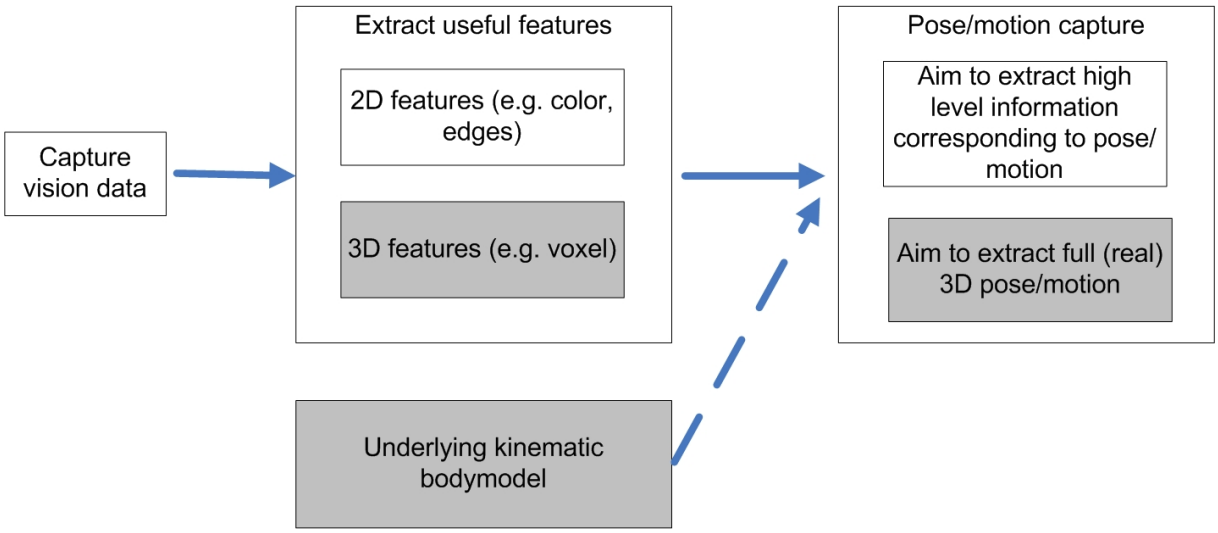


Figure 5.2: Block diagram of a generic human body pose estimation system. Dash line means that the underlying kinematic model can be used or not. Gray boxes show the focus of this paper, which are model based methods using voxel data and aim to extract full 3D posture.

and the environment is static, there are still remaining key challenges including tracking with fewer cameras (< 4), dealing with complex environments, variations in object appearance (e.g. general clothes, hair, etc.), automatically adapting to different body shapes, and automatically recovering from failure.

Some surveys of several techniques for human body pose modeling and tracking can be found in [56, 57, 67, 100], each with different focus and taxonomy. Werghi [100] provided a general overview of both 3D human body scanner technologies and approaches dealing with such scanned data, which focus on one or more of the following topics: body landmark detection, segmentation of body scanned data, body modeling and body tracking. Poppe [67] survey on pose estimation techniques, in which they mentioned the division into 2D approaches and 3D approaches, depends on the goal to achieve 2D pose or 3D pose representation; The division into model-based approaches and model-free approaches, depends on whether a priori kinematic body model is employed. Moeslund et al. [56] split the pose estimation process into initialization, tracking, pose estimation, and recognition. In [57], they also provided an updated review of advances in human motion capture for the period from 2000 to 2006. We see that it is not easy to have a unified taxonomy for the broad area of human body modeling and tracking. Quite similar to [56], we categorize related research studies based on the common components in a generic body pose estimation system. As shown in Fig. 5.2, we first need a component to extract useful features from the input vision data, and then a procedure to infer body pose from extracted features. In this paper, we focus on representative model-based approaches using multi-view video input and aim to extract real 3D posture. In comparison to monocular view, multi-view data can help to reduce the self occlusion issue and provide more information to make the pose estimation task easier as well as to improve the accuracy. The underlying kinematic body model in model-based approaches can help to improve the accuracy and robustness although it also raises the issue of model initialization and re-initialization.

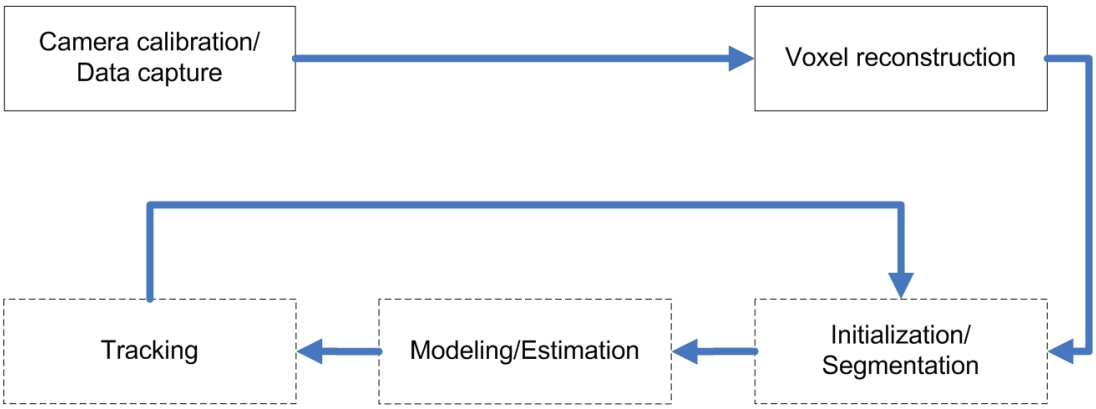


Figure 5.3: Common steps in model-based methods for articulated human body pose estimation using multi-view input. Dashed boxes mean that some methods mayor may not have all of these steps.

5.1.2 Human Action Recognition

While 2D human action recognition has received high interest during the last decade, 3D human action recognition is still a less explored field. Relatively few authors have so far reported work on 3D human action recognition [39, 57, 68, 99]. Human actions are performed in real 3D environments, however, traditional cameras only capture the 2D projection of the scene. Vision-based analysis of 2D activities carried out in the image plane will therefore only be a projection of the actual actions. As a result, the projection of the actions will depend on the viewpoint, and not contain full information about the performed activities. To overcome this shortcoming the use of 3D representations of reconstructed 3D data has been introduced through the use of two or more cameras [1, 27, 37, 74, 96]. In this way the surface structure or a 3D volume of the person can be reconstructed, e.g., by Shape-from-Silhouette (SfS) techniques [79], and thereby a more descriptive representation for action recognition can be established.

The use of 3D data allows for efficient analysis of 3D human activities. However, we are still faced with the problem that the orientation of the subject in the 3D space should be known. Therefore, approaches have been proposed without this assumption by introducing view-invariant or view-independent representations. Another strategy which has been explored is the application of multiple views of a scene to improve recognition by extracting features from different 2D image views or to achieve view-invariance.

The ultimate goal is to be able to perform reliable action recognition applicable for, e.g., video annotation, advanced human computer interaction, video surveillance, driver assistance, automatic activity analysis and behavior understanding. We contribute to this field by providing a review and comparative study of recent research on 2D and 3D human action recognition for multi-view camera systems (see Table 5.5 and 5.6), to give people interested in the field an easy overview of the proposed approaches, and an idea of the performance and direction of the research.

Methods for 3D human action recognition can either be model-free or model-based. Mostly a model-free strategy is applied, which have the advantage that it can use a wide range of image, static shape/pose, motion or statistical features, and does not depend on a

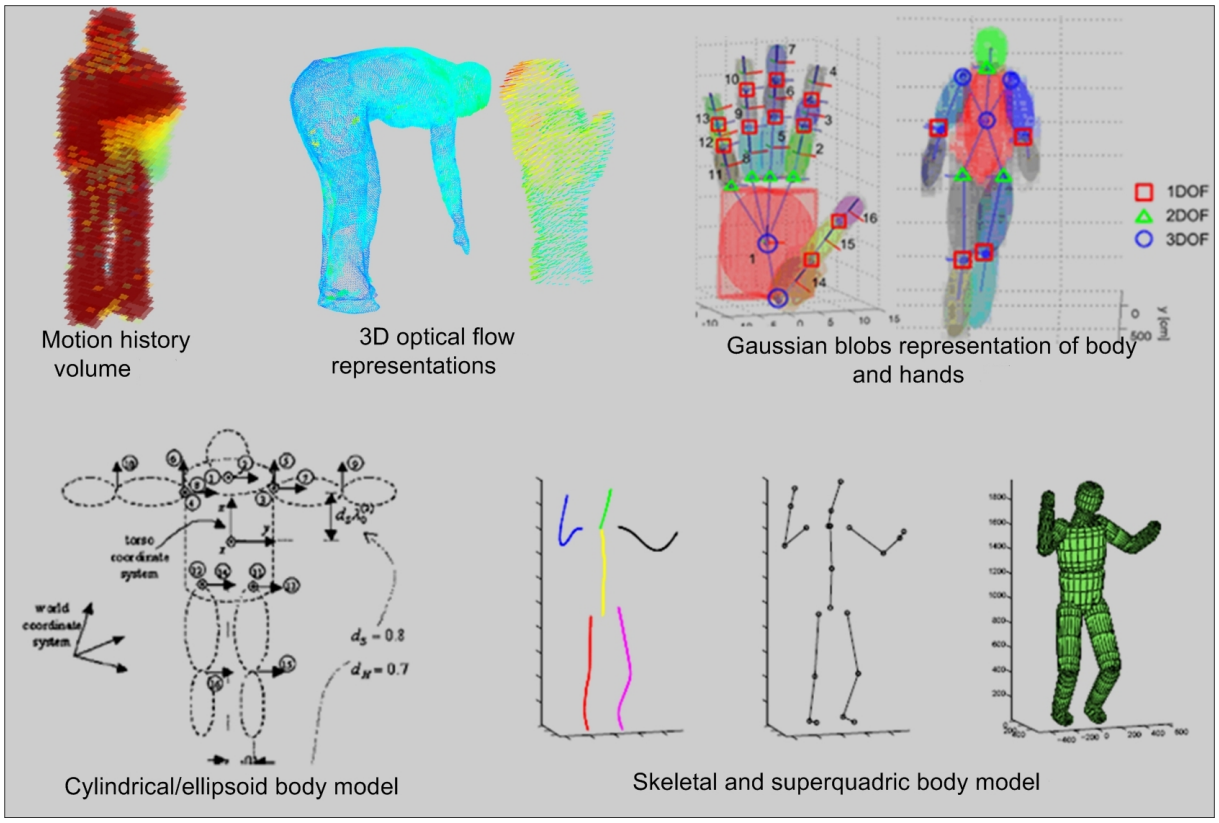


Figure 5.4: Prominent 3D human body model and human motion representations [13, 32, 55, 81, 96].

predefined human body model. However, the approaches usually do not capture any information about the 3D human body pose, joint positions etc. This limits it's usability to a specific set of applications, where the exact pose and joint configuration of the body parts are not explicitly required (see Fig 5.1 and Table 5.1). Whereas, the model-based methods, which requires a human body model and is usually applied in conjunction with human body modeling and pose estimation, allows for description of the exact pose of the respective body parts. This opens up for another set of applications.

The remainder of the paper is organized as follows. Section 5.2 is a review of selected recent model-based methods for human body pose estimation using multi-view data. Section 5.3 gives a review of 2D and 3D approaches for human action recognition, followed up by a description of multi-view dataset and a quantitative comparison of promising methods. Finally in Section 5.4, we present a discussion and directions of future work.

5.2 3D Human Body Modeling and Pose Estimation

As mentioned earlier in Section 5.1.1, in this paper we focus on model-based approaches using multi-view video input and aim to extract real 3D posture. Figure 5.3 shows common steps in model-based approaches for human pose estimation using multi-view input including: camera calibration/data capture, voxel reconstruction, initialization/segmentation

(segment voxel data into different body parts), modeling/estimation (estimating pose using the current frame only), and tracking (use temporal information from the previous frames in estimating body pose in the current frame). In each step, different methods may have different choices of approaches: There are methods using 3D features (e.g. voxel data) reconstructed from multiple views while others may still use 2D features (e.g. silhouette, contour) extracted from each view. They may have manual or automatic initialization step. Some methods may not have tracking step. Some methods are for a generic purpose while others are application specific for efficiency. Table 5.2, 5.3 and 5.4 are a summary of selected representative model based methods for human body pose estimation using multi-view data (see Fig. 5.4). In the following section, we will discuss in more details the factors mentioned above.

5.2.1 Using 2D vs. 3D features from multi-view

Among multi-view approaches, some methods use 3D features reconstructed from multiple views [9, 12, 13, 16, 17, 20, 55, 81], e.g. volumetric (voxel) data, while others still use 2D features [36, 45, 66], e.g. color, edges, silhouette. Since the real body pose is in 3D, using voxel data can help avoiding the repeated projection of 3D body model onto the image planes to compare against the extracted 2D features. Furthermore, reconstructed voxel data help to avoid the image scale issue. These advantages of using voxel data allow the design of simple algorithms and we can make use of our knowledge about shapes and sizes of body parts. For example, Mikic et al. [55] used specific information about shape and size of head and torso to have a hierarchical growing procedure (detecting head first, then torso, then limbs) for body model acquisition that can be used effectively even when there is a large displacements between frames. Several methods are based on voxel data, which only indicates that voxel data is a strong cure for body pose estimation. Of course, there is an additional computational cost for voxel reconstruction but efficient techniques for this task have also been developed [9, 17, 16, 76].

5.2.2 Tracking-based vs. single frame-based approaches

The modeling and tracking steps can be considered as a mapping from input space of voxel data Y and information in the predefined model (e.g. kinematic constraints) C to the body model configuration space Θ:

$$M : (Y, C) \mapsto \Theta \tag{5.1}$$

The body model configuration contains both static parameters (i.e. shape and size of each body component) and dynamic parameters (i.e. mean and orientation of each body component), in which the static parameters are estimated in the initialization step. Methods are different in the way they use and implement the mapping procedure M. Methods that have modeling step but no tracking step are also called single frame-based methods, e.g. [81] while methods with tracking step are called tracking-based methods, e.g. [9, 13, 19, 25, 49, 55, 86]. Because the tracker in tracking based methods would be lost over long sequences, multiple hypotheses at each frame can be used to improve the robust-

Table 5.2: Summary of selected model-based methods for multi-view body pose estimation and tracking.

Year	First author	Data acquisition	Initialization	Body model	Method highlights	Evaluation & Precision	Real-time
2001	Delamarre [20]	3 cameras	Manual	Truncated cones, spheres, and parallelepipeds 3D body model	Uses physical forces, a simpler form of Iterative Closest Point (ICP) to track 3D body model from voxel data. Kalman Filter tracking	Visual only (qualitative)	N/A
2003	Mikic [55]	6 cameras	Automatic	Ellipsoidal, cylindrical 3D body model	Hierarchically growing procedure for initialization from head to torso, to limbs. Uses Extended Kalman Filter to predict next pose then update using growing procedure and Bayesian networks	Visual only (qualitative)	N/A
2003	Cheung [16]	8 cameras	Automatic	Skeletal body model	Uses Colored Surface Point (CSP). Hierarchical segmentation & SFS alignment to recover motion, shape and joint	Synthesized ground-truth. Average joint position error $\sim$ 2cm	N/A
2006	Ziegler [102]	4 cameras	Manual	3D skeletal upper body model	Reconstruct 3D voxel data. Using ICP algorithm integrated with an Unscented Kalman Filter (UKF) to track 3D upper body motion	Manual label ground-truth. Joint angle error $\sim 20^0$	1 fps (frame per sec)
2007	Cheng [13]	4 cameras	Manual	Ellipsoidal 3D body model & Gaussian representation	Integrate kinematic constraints in Kinematically Constrained Gaussian Mixture Model (KC-GMM). Derive EM algorithm with KC-GMM for pose estimation (no additional projection step)	Joint position error $\sim$0.5cm for synthesized hand data, $\sim$17cm for HumanEvaII body data	N/A

Table 5.3: Summary of selected model-based methods for multi-view body pose estimation and tracking.

Year	First author	Data acquisition	Initialization	Body model	Method highlights	Evaluation & Precision	Real-time
2008	Caillette [9]	3-5 cameras	Manual	Skeletal body model & Gaussian blobs	Break complex movement into basic motions Use Variable Length Markov Model (VLMM) to predict candidate pose. Use colored voxel for more robust tracking. Limited to a prior training motion model	Joint position error ∼2cm for a reported sequence	5-28 fps
2008	Sundaresan [81]	4-12 cameras	Automatic	6-chains representation and superquadric 3D body model	Segment voxel data in Laplacian Eigenspace (LE). Probabilistic register segmented voxel to body parts then estimate skeletal and superquadric parameters. No tracking	Synthesized data (joint angle error $\sim 5^0$, non-public dataset, and HumanEvaII (loss of track)	N/A
2009	Tran [86]	2 cameras (wide baseline)	Automatic	3D skeletal upper body model	Track head and head blobs. Only use 3D head and hands movements to infer corresponding the whole upper body movement as an inverse kinematics problem	Non-public dataset and HumanEvaI. Joint position error ∼10cm	15 fps
2009	Bernier [8]	Stereo	Automatic	3D skeletal upper body model	Use a graphical model to decompose the full 3D pose state space into individual limb state space, coupled with a fast Nonparametric Belief Propagation for articulated pose tracking	Synthesized data. Joint position error ∼7cm	10 fps

Table 5.4: Summary of selected model-based methods for multi-view body pose estimation and tracking.

Year	First author	Data acquisition	Initialization	Body model	Method highlights	Evaluation & Precision	Real-time
2010	Gall [25]	4 cameras	Manual	3D surface mesh model	Multi-layer framework combining global optimization, smoothing, and local optimization to improve silhouette segmentations. Track 3D pose with ICP	HumanEvaII dataset. Joint position error ∼5cm	N/A
2010	Corazza [19]	4-12 cameras	Automatic	3D surface mesh model	Automatic pose-shape registration based on a database of human body shapes. Use ICP for 3D pose tracking	HumanEvaII and non-public dataset. Joint position error 1.5 - 8cm	N/A
2010	Li [49]	3 cameras	Manual	Truncated cone 3D body model	Learn a low-dimensional manifold of human body pose from motion capture data using coordinated mixture of factor analyzer. Use Bayesian framework to track 3D human body	HumanEvaI dataset. Joint position error ∼7cm	N/A
2011	Hofmann [31]	3 cameras	Automatic	Super-quadrics 3D body model	Single-frame pose recovery with 2D pose exemplar generation; Select 3D pose candidates with Bayesian framework; Refine with temporal integration and model texture adaptation.	HumanEvaI and non-public dataset in complex environment. Joint position error ∼10cm	N/A

ness of tracking. Single frame based approach is a more difficult issue because it does not make any assumptions on time coherence. However, we see that tracking-based methods encounter the issue of initialization or re-initialization of the tracked model.

5.2.3 Manual vs. automatic initialization

Some methods have automatic initialization step like [16, 31, 55, 81, 86] while others require a priori known or manually initialized static parameters, e.g. [13, 20, 25, 49, 102]. In [55], the specific shape and size of the head was used to design a hierarchical growing procedure for initialization. In [19], a database of human body shapes was used for initial pose-shape registration. In [8, 86], the user was asked to start at a specific pose (e.g. stretch pose) to aid the automatic initialization. In [81], Sundaresan et al. discovered an interesting property of Laplacian Eigenspace (LE) transformation: By mapping into high dimensional (e.g. 6D) LE, voxel data of body chains like limbs, which have their length greater than their thickness, will form an 1-D smooth curve which can then be used to segment voxel data into different body chains. They then use a spline fitting process to segment the curves which results in the segmentation of their respective body chains. This is however a single frame based approach: The segmented voxel clusters are then registers to their actual body chain using a probabilistic registration procedure at each frame. Their results seem to be sensitive to noise in the voxel data (loss of track in the test with the public HumanEvaII dataset).

On the other hand, the Kinematically Constrained Gaussian Mixture Model (KC-GMM) method proposed by Cheng and Trivedi [13] is a tracking based method and showed good results on the HumanEvaII dataset (won the first prize in the Workshop on Evaluation of Articulated Human Motion and Pose Estimation - CVPR EHuM2 2007 competition). However it requires a careful manual initialization. An framework combining KC-GMM method and LE-based voxel segmentation was proposed in [83] for a more powerful human body modeling and tracking system. The LE based voxel segmentation was used to fill in the gap of an automatic initialization of KC-GMM method. Regarding the LE-based method, combining with a tracking based method like KC-GMM instead of doing voxel segmentation at every frame helps to overcome the sensitization to voxel noise to some extent.

5.2.4 Generic purpose vs. application specific approaches for efficiency

Depending on applications, human pose tracking may focus on different body parts including full body pose, upper body pose [57, 67], hand pose [21], and head pose [60]. Due to the complexity of human body pose estimation task, there are trade-offs between developing a generic approach versus an approach integrated to some specific cases for efficiency. For example, the KC-GMM method [13] is for generic purpose and was applied successfully for both HumanEvaII body data and synthesized hand data. However, this method is not real-time because of a required manual initialization step and related computational cost. For efficiency, some methods are designed for application specific. For

example [86, 8] focus on situations in which most of the influential information of body motion carried by the upper body and arms while the user typically in a fixed position. These situations arise in several realistic applications such as driver activity analysis and user activity analysis in a smart teleconference or meeting room. In [86], the problem of upper body pose tracking is broken into two sub-problems: First track the extremities including head and hands blobs. Then the 3D movements of head and hands are used to infer the corresponding upper body movements as an inverse kinematics problem. Since the head and hand regions are typically well defined and undergo less occlusion, tracking is more reliable. Moreover by breaking the high dimensional search problem of upper body pose tracking into two sub-problems, the complexity is reduced considerably to achieve real-time performance. However they need to deal with possible ambiguity due the kinematic redundancy of body model.

Another type of approaches for efficiency is to use a prior motion model from training sequences. Some representative approaches using prior motion models are [9] learning prior motion model with Variable Length Markov Model (VLMM), which can explain high-level behaviors over a long history or [49] using coordinated mixture of factor analyzer to learn the prior model. Compared to approaches for generic body motions [13, 19, 25, 31, 55], these approaches use the prior motion models to reduce the search space for a more efficient and robust pose tracking. However the downside is that these methods are limited to the type of motions in training data (i.e. have difficulties if there are "unseen" movements)

5.3 Multi-View Human Action Recognition

In this section we review and compare multi-view approaches for human action recognition (see Table 5.5, 5.6 and 5.7). First we will give an outline of approaches which solely apply 2D multi-view image data, then full 3D-based techniques, followed up by a description of publicly available multi-view datasets and a comparison of several promising methods based on evaluations on the INRIA Xmas Motion Acquisition Sequences (IXMAS) Multi-View Human Action Dataset [96] and the i3DPost Multi-View Human Action and Interaction Dataset [27].

5.3.1 2D Approaches

One line of work concentrates solely on the 2D image data acquired by multiple cameras. Action recognition can range from pointing gesture to complex multi-signal actions, e.g., including both coarse level of body movement and fine level of hand gesture. Matikainen et al. [54] proposed a method for multi-user, prop-free pointing detection using two camera views. The observed motion are analyzed and used to refer the candidates of pointing rotation centers and then estimate the 2D pointer configurations in each image. Based on the extrinsic camera parameters, these 2D pointer configurations are merged across views to obtain 3D pointing vectors.

In the work of Souvenir et al. [78], the acquired data from 5 calibrated and synchronized cameras, is further projected to 64 evenly spaced virtual cameras used for training. Ac-

Table 5.5: Publications on multi-view human action recognition.

Year	First author	Dim	Feature/Representation	Classifier/Matching	Other techniques	Comments
2005	K. Huang [33]	3D	3D shape context	Hidden Markov model Maximum likelihood	Tracking	Real-time (∼10 fps) Multiple people
2006	Ahmad [2]	2D	Optical flow and PCA Human body shape	Hidden Markov model Maximum likelihood		Single person Training from multi-view
2006	Canton-Ferrer [10]	3D	3D Motion descriptors 3D invariant statistical moments	Bayesian classifier	Ellipsoid body model Tracking	Multiple people
2006	Pierobon [65]	3D	Cylindrical shape descriptor	Template matching	Dynamic time warping	Possible real-time Single person
2006	Weinland [96]	3D	Motion history Volumes Cylindrical Fourier transform	Mahalanobis distance	PCA LDA	Single person
2007	Lv [53]	2D	Shape context of 2D poses graph model: Action Net	Pyramid Match Kernel Viterbi algorithm		Near real-time Single person Synthetic training data
2007	Weinland [98]	2D	3D exemplars Silhouette projections	Hidden Markov model Maximum a posteriori		3D learning 2D classification Single person
2008	Cherla [15]	2D	Width profile, eigenanalysis Spatio-temporal features	Average template matching	Dynamic time warping temporal discriminative weighting	Real-time (∼20 fps) Two orthogonal views Single person
2008	Farhadi [22]	2D	Histogram of silhouette and optic flow	Nearest Neighbor Hamming distance	Transferable activity model Vector quantization	Cross-view recognition Single person

Table 5.6: Publications on multi-view human action recognition.

Year	First author	Dim	Feature/Representation	Classifier/Matching	Other techniques	Comments
2008	Junejo [41]	2D	Self-Similarity Matrix (SSM) Bag of local SSM descriptors	Support vector machines Nearest neighbor		Cross-view recognition Single person
2008	Liu [50]	2D	Local spatio-temporal volumes Spin-images	Fiedler Embedding	Graph-based Multiple features	Single person
2008	Liu [51]	2D	Spatio-temporal interest points Bag of Cuboid features	Support vector machines	Maximum mutual info. Structural information Compressed codebook	Single person
2008	Souvenir [78]	2D	$\mathcal{R}$ transform surfaces Manifold learning	2D diffusion distance		64 virtual camera views Single person
2008	D. Tran [87]	2D	Motion context	Nearest neighbor Naive Bayes	Metric learning	Reject unfamiliar samples Learn from few examples Single person
2008	Turaga [92]	3D	Motion history volumes Stiefel and Grassmann manifolds	Procrustes distance metric	Statistical modeling	Single person Applicable for other applications
2008	Vitaladevuni [94]	2D	Ballistic dynamics	Bayesian model	Motion history image	Single person
2008	Yan [101]	3D	Spatio-Temporal Volumes (STV) 4D action feature model	Maximum likelihood function	Local STV features	Single person
2009	Gkalelis [28]	2D	Multi-view posture masks Discrete Fourier transform	Mahalanobis distance	Fuzzy vector quantization LDA	Single person
2009	Kilner [44]	3D	Shape similarity	Markov model	Tracking	Outdoor sports broadcasts Multiple people

Table 5.7: Publications on multi-view human action recognition.

Year	First author	Dim	Feature/Representation	Classifier/Matching	Other techniques	Comments
2009	Reddy [71]	2D	Feature-tree of Cuboids	Local voting strategy		Near real-time Localization of the action Multiple people
2010	P. Huang [35]	3D	Shape-flow descriptor Shape histogram	Similarity matrix	Time-filtering	Comparison of descriptors Single person
2010	Iosifidis [38]	2D	Multi-view binary masks	Euclidean distances Mahalanobis distance	Fuzzy vector quantization LDA	Single person
2010	Weinland [97]	2D	3D Histogram of Oriented Gradients (HOG)	Hierarchical classification Local Support vector machines		Near real-time Local occlusion handling Single person
2011	Haq [30]	2D	Dynamic scene geometry	Multi-body fundamental matrix	Epipolar geometry	Action retrival Single person
2011	Holte [32]	3D	3D optical flow Harmonic motion context	Normalized correlation	Spherical harmonics	Highly detailed 3D motion Single person
2011	Junejo [42]	2D	Temporal self-similarities Bag of SSM descriptors	Support vector machines Nearest neighbor	Dynamic time warping	Cross-view recognition Single person
2011	Liu [52]	2D	Bag of bilingual words Cuboid features	Graph matching	View knowledge transfer	Cross-view recognition Single person
2011	Pehlivan [64]	3D	Pose descriptors Circular body layer features	Nearest neighbor	Combining pose descriptors into motion matrices	Single person
2011	Song [77]	3D	3D Pose and HOG features	Hidden conditional random fields		Body and hand movements Single person

tions are described in a view-invariant manner by computing $\mathcal{R}$ transform surfaces of silhouettes and manifold learning. Gkalelis et al. [28] exploits the circular shift invariance property of the Discrete Fourier Transform (DFT) magnitudes, and use Fuzzy Vector Quantization (FVQ) and Linear Discriminant Analysis (LDA) to represent and classify actions. Another approach was proposed by Iosifidis et al. [38], where Binary body masks from frames of a multi-camera setup used to produce the i3DPost Multi-View Human Action Dataset [27], are concatenated to multi-view binary masks. These masks are rescaled and vectorized to create feature vectors in the input space. FVQ is performed to associate input feature vectors with movement representations and LDA is used to map movements in a low dimensionality discriminant feature space.

Some authors perform action recognition from image sequences in different viewing angles. Ahmad et al. [2] apply Principal Component Analysis (PCA) of optical flow velocity and human body shape information, and then represent each action using a set of multi-dimensional discrete Hidden Markov Models (HMM) for each action and viewpoint. Cherla et al. [15] show how view-invariant recognition can be performed by using data fusion of two orthogonal views. An action basis is built using eigenanalysis of walking sequences of different people, and projections of the width profile of the actor and spatio-temporal features are applied. Finally, Dynamic Time Warping (DTW) is used for recognition. A number of other techniques have been employed, like metric learning [87] or representing action by feature-trees [71] or ballistic dynamics [94]. In [97] Weinland et al. propose an approach which is robust to occlusions and viewpoint changes using local partitioning and hierarchical classification of 3D Histogram of Oriented Gradients (3DHOG) volumes.

Others use synthetic data rendered from a wide range of viewpoints to train their model and then classify actions in a single view, e.g. Lv et al. [53], where shape context is applied to represent key poses from silhouettes and Viterbi Path Searching for classification. A similar approach was proposed by Fihl. et al. [23] for gait analysis.

Another topic which has been explored by several authors the last couple of years is cross-view action recognition. This is a difficult task of recognizing actions by training on one view and testing on another completely different view (e.g., the side view versus the top view of a person in IXMAS). A number of techniques have been proposed, stretching from applying multiple features [50], information maximization [51], dynamic scene geometry [30], self similarities [41, 42] and transfer learning [22, 52]. For additional related work on view-invariant approaches please refer to the recent survey by Ji et al. [39].

5.3.2 3D Approaches

Another line of work utilize the full reconstructed 3D data for feature extraction and description. Figure 5.4 shows some examples of the more prominent model and non-model-based representations of the human body and its motion. These will be reviewed in the following along with a number of other recent 3D approaches.

Johnson and Hebert proposed the spin image [40], and Osada et al. the shape distribution [62]. Ankerst et al. introduced the shape histogram [5], which is a similar to the 3D extended shape context [7] presented by Körtgen et al. [47], and Kazhdan et al. applied

spherical harmonics to represent the shape histogram in a view-invariant manner [43]. Later Huang et al. extended the shape histogram with color information [34]. Recently, Huang et al. made a comparison of these shape descriptors combined with self similarities, with the shape histogram (3D shape context) as the top performing descriptor [35].

A common characteristic of all these approaches is that they are solely based on static features, like shape and pose description, while the most popular and best performing 2D image descriptors apply motion information or a combination of the two [57, 99]. Some authors add temporal information by capturing the evolvement of static descriptors over time, i.e., shape and pose changes [10, 18, 33, 44, 65, 96, 98, 101]. The common trends are to accumulate static descriptors over time, track human shape or pose information, or apply sliding windows to capture the temporal contents [57, 65, 96, 99]. Cohen et al. [18] use 3D human body shapes and Support Vector Machines (SVM) for view-invariant identification of human body postures. They apply a cylindrical histogram and compute an invariant measure of the distribution of reconstructed voxels, which later was used by Pierobon et al. [65] for human action recognition. Another example is seen in the work of Huang and Trivedi [33], where a 3D cylindrical shape context is presented to capture the human body configuration for gesture analysis of volumetric data. The temporal information of an action is modeled using HMM. However, this study does not address the view-independence aspect. Instead, the subjects are asked to rotate while training the system.

More detailed 3D pose information (i.e. from tracking the kinematics model of the human body) is a rich and view-invariant representation for action recognition but challenging to derive [68]. Human body pose tracking is itself an important area with many related research studies. Among these, research started with monocular view and 2D features, and more recently (about 10 years ago) multi-view and 3D features like volumetric data have been applied for body pose estimation and tracking [84]. One of the earliest methods for multi-view 3D human pose tracking using volume data was proposed by Mikic et al. [55], in which they use a hierarchical procedure starting by locating the head using its specific shape and size, and then growing to other body parts. Though this method showed good visual results for several complex motion sequences, it is also quite computationally expensive. Cheng and Trivedi [13] proposed a method that incorporates the kinematics constraints of a human body model into a Gaussian Mixture Model framework, which was applied to track both body and hand models from volume data. Although this method was highly rated with good body tracking accuracy on HumanEva dataset [74], it requires a manual initialization and could not run in real-time. We see that there are always trade-offs between achieving detailed information of human body pose and the computational cost as well as the robustness. In [77], Song et al. focus on gestures with more limited body movements. Therefore they only use the depth information from two camera views to track 3D upper body poses using a Bayesian inference framework with a particle filter, as well as classifying several hand poses based on their appearance. The temporal information of both upper body and hand pose are then inputted into a Hidden Conditional Random Field (HCRF) framework for aircraft handling gesture recognition. To deal with the long range temporal dependencies in some gestures, they also incorporate a Gaussian temporal smoothing kernel into the HCRF inference framework.

The Motion History Volume (MHV) was proposed by Weinland et al. [96], as a 3D exten-

sion of Motion History Images (MHIs) (see Fig. 5.4). MHVs are created by accumulating static human postures over time in a cylindrical representation, which is made view-invariant with respect to the vertical axis by applying the Fourier transform in cylindrical coordinates. The same representation was used by Turaga et al. [92] in combination with a more sophisticated action learning and classification based on Stiefel and Grassmann manifolds. Later, Weinland et al. [98] proposed a framework, where actions are modeled using 3D occupancy grids, built from multiple viewpoints, in an exemplar-based Hidden Markov Models (HMM). Learned 3D exemplars are used to produce 2D image information which is compared to the observations, hence, 3D reconstruction is not required during the recognition phase.

Pehlivan et al. [64] presented a view-independent representation based on human poses. The volume of the human body is first divided into a sequence of horizontal layers, and then the intersections of the body segments with each layer are coded with enclosing circles. The circular features in all layers: (i) the number of circles, (ii) the area of the outer circle, and (iii) the area of the inner circle are then used to generate a pose descriptor. The pose descriptors of all frames in an action sequence are further combined to generate corresponding motion descriptors. Action recognition is then performed with a simple nearest neighbor classifier.

Canton-Ferrer et al. [10] propose another view-invariant representation based on 3D MHIs and 3D invariant statistical moments. Recently, Huang et al. proposed 3D shape matching in temporal sequences by time filtering and shape flows [35]. Kilner et al. [44] applied the shape histogram and evaluated similarity measures for action matching and key-pose detection in sports events, using 3D data available in the multi-camera broadcast environment. A different strategy is presented by Yan et al. [101]. They propose a 4D action feature model (4D-AFM) for recognizing actions from arbitrary views based on spatio-temporal features of spatio-temporal volumes (STVs). The extracted features are mapped from the STVs to a sequence of reconstructed 3D visual hulls over time, resulting in the 4D-AFM model, which is used for matching actions.

A 3D descriptors which are directly based on rich detailed motion information are the 3D Motion Context (3D-MC) [32] and the Harmonic Motion Context (HMC) [32] proposed by Holte et al. The 3D-MC descriptor is a motion oriented 3D version of the shape context [7, 47], which incorporates motion information implicitly by representing estimated 3D optical flow (see Fig. 5.4) by embedded Histograms of 3D Optical Flow (3D-HOF) in a spherical histogram. The HMC descriptor is an extended version of the 3D-MC descriptor that makes it view-invariant by decomposing the representation into a set of spherical harmonic basis functions.

5.3.3 Multi-View Datasets

A number of multi-view human action datasets are publicly available. A frequently used dataset is the INRIA Xmas Motion Acquisition Sequences (IXMAS) Multi-View Human Action Dataset[1] [96]. It consists of 12 non-professional actors performing 13 daily-life

[1]The IXMAS dataset is available at http://4drepository.inrialpes.fr/public/viewgroup/6

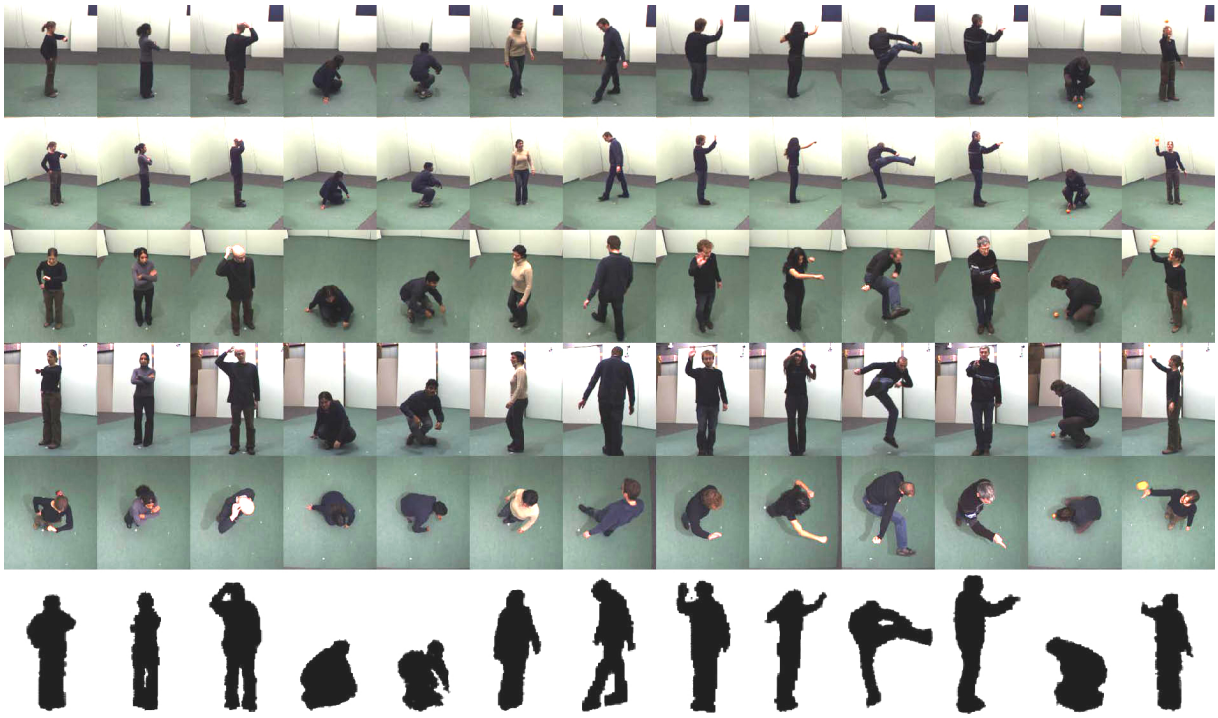


Figure 5.5: Image and 3D voxel-based volume examples for the 13 actions from the IXMAS Multi-View Human Action Dataset. The figure is organized such that the columns correspond to the 13 different actions performed by the 12 actors. The first 5 rows depict images captured from the 5 camera views, while the 6^{th} row shows the corresponding 3D volumes.

actions 3 times: *check watch*, *cross arms*, *scratch head*, *sit down*, *get up*, *turn around*, *walk*, *wave*, *punch*, *kick*, *point*, *pick up* and *throw*. The dataset has been recorded by 5 calibrated and synchronized cameras, where the actors chose freely position and orientation, and consists of image sequences (390×291) and reconstructed 3D volumes ($64 \times 64 \times 64$ voxels), resulting in a total of 2340 action instances for all 5 cameras. Figure 6.9 shows multi-view actor/action images and voxel-based volume examples from the IXMAS datasets.

Recently, a new high quality dataset has been produced, the i3DPost Multi-View Human Action and Interaction Da-taset[2] [27]. This dataset, which has been generated within the Intelligent 3D Content Extraction and Manipulation for Film and Games EU funded research project, consists of 8 actors performing 10 different actions, where 6 are single actions: *walk*, *run*, *jump*, *bend*, *hand-wave* and *jump-in-place*, and 4 are combined actions: *sit-stand-up*, *run-fall*, *walk-sit* and *run-jump-walk*. Additionally, the dataset also contains 2 interactions: *handshake* amd *pull*, and 6 basic facial expressions. The subjects have different body sizes, clothing and are of different sex and nationalities. The multi-view videos have been recorded by a 8 calibrated and synchronized camera setup in high definition resolution (1920×1080), resulting in a total of 640 videos (excluding videos of interactions and facial expressions). For each video frame a 3D mesh model of relatively high detail level ($20,000$-$40,000$ vertices and $40,000$-$80,000$ triangles) of the actor and

[2]The i3DPost dataset is available at http://kahlan.eps.surrey.ac.uk/i3dpost_action/data

Figure 5.6: Image and 3D mesh model examples for the 10 actions from the i3DPost Multi-View Human Action Dataset. The figure is organized such that the columns correspond to the 10 different actions performed by the 8 actors, where the first 6 columns show the single actions and the last 4 columns show the combined actions. The first 8 rows depict images captured from the 8 camera views, while the 9^{th} row shows the corresponding 3D mesh models.

Table 5.8: Recognition accuracies (%) for the IXMAS dataset. The column named "Dim" states if the methods apply 2D image data or 3D data, the other columns states how many actions are used for evaluation, and if the results are based on all views or cross-view recognition.

Year	Method	Dim	11 actions	13 actions	All views	Cross-view
2008	Turaga et al. [92]	3D	98.78	-	x	
2006	Weinland et al. [96]	3D	93.33	-	x	
2011	Pehlivan et al. [64]	3D	90.91	88.63	x	
2008	Vitaladevuni et al. [94]	2D	87.00	-	x	
2011	Haq et al. [30]	2D	83.69	-		x
2010	Weinland et al. [97]	2D	83.50	-	x	
2008	Liu et al. [51]	2D	-	82.80	x	
2011	Liu et al. [52]	2D	82.80	-		x
2007	Weinland et al. [98]	2D	81.27	-	x	
2007	Lv et al. [53]	2D	-	80.60	x	
2008	Tran et al. [87]	2D	-	80.22	x	
2008	Cherla et al. [15]	2D	-	80.05	x	
2008	Liu et al. [50]	2D	-	78.50	x	
2008	Yan et al. [101]	3D	78.00	-	x	
2011	Junejo et al. [42]	2D	74.60	-	x	
2008	Junejo et al. [41]	2D	72.70	-	x	
2009	Reddy et al. [71]	2D	-	72.60	x	
2008	Farhadi et al. [22]	2D	58.10	-		x

the associated camera calibration parameters are available. The mesh models were reconstructed using a global optimization method proposed by Starck and Hilton [79]. Figure 6.8 shows multi-view actor/action images and 3D mesh model examples from the i3DPost dataset.

Another interesting multi-view dataset is the Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion (HumanEva) [74], containing 6 simple actions performed by 4 actors, captured by 7 calibrated video cameras (4 grayscale and 3 color), which have been synchronized with 3D body poses obtained from a motion capture system. Among other less frequently used multi-view datasets are the CMU Motion of Body (MoBo) Database [29], the Multi-camera Human Action Video Data-set (MuHAVi) [1] and the KU Gesture Dataset [37].

5.3.4 Comparison

In this section we report a quantitative comparison of the reviewed approaches using two publicly available datasets: IXMAS and i3DPost.

In Table 6.2 the recognition accuracies of several 2D and 3D approaches evaluated on IXMAS are listed. It is interesting to note that all the 3D approaches except one are the top performing methods. This indicates that the use of the full reconstructed 3D

Table 5.9: Recognition accuracies (%) for the i3DPost dataset. *Gkalelis et al. [28] test on 5 single actions.

Year	Method	Dim	8 actions
2011	Holte et al. [32]	3D	92.19
2010	Iosifidis et al. [38]	2D	90.88
2009	Gkalelis et al. [28]	2D	90.00*

information is superior to applying 2D image data from multiple views, when it comes to recognition accuracy. However, the computational cost of working in 3D is usually also more expensive. Hence, with respect to the application and demand for real-time performance, 2D approaches might still be best choice. It should be noted that some results are reported using cross-view evaluation, which is more challenging than applying data from multiple and identical viewpoints, however, still some of these methods perform very well. When both types of results are available in the original work, we have reported the results for all views, since these are more comparable to the 3D Results, where all views are used to reconstruct 3D data.

Table 6.1 shows the recognition accuracies of a few other approaches evaluated on the i3DPost dataset. The evaluation has been carried out for 8 actions by combining the 6 single actions in the dataset with two additional single actions: *sit down* and *fall* by splitting 2 of the 4 combined actions. Again the approach based on full 3D information outperforms the 2D methods.

The top performing approaches for the two datasets are the 3D-based methods by Turaga et al. [92], Weinland et al. [96] and Holte et al. [32], Where both [92] and [96] are based on Motion History Volumes (MHVs), and [32] are based on 3D optical flow and Harmonic Motion Context (HMC). However, it should be noticed that all these methods for human action recognition are basically model-free, which means that they do not apply a specific human body model to model and estimate the exact position and configuration of the body parts and joints. Hence, these methods are only applicable for a set of the application in Table 5.1. This results in a need for model-based approaches for 3D pose estimation and exact modeling of the human body.

5.4 Discussion and Future Directions

In this paper, we provide a review and comparative study of recent developments for human body modeling, pose estimation and activity recognition using multi-view data. we give a overview of the different application areas and their associated requirements for successful operation.

We provide a review of the sub-area of model-based method for real human body pose estimation using volumetric data. After a brief overview to put in context this concerned subarea, we focus on analyzing and comparing several selected methods, especially some recent methods in the past two years to high light their important results including increasing generality, real time performance, and a new general LE based method for voxel segmentation. Based on this analysis, we discuss about our idea of a method combin-

ing LE based voxel segmentation and KC-GMM methods for an automatic human body model initialization and tracking using voxel data. A close follow up work for us is to implement this idea. We may think of several other directions for future work in improving performance and robustness of current pose estimation methods. First, we can keep trying to combine good characteristics from different methods to have a more robust one. For example, we may want to incorporate some kind of prediction information as done in [9, 55] to the proposed combined method. Second, we can find some way to use both 3D voxel feature and 2D features. In [9, 16] they have associate color information to voxel data. Other 2D features like edges, appearance model, etc should be also useful. Regarding the major difficulty of high-dimensional body pose configuration space, we can also exploit the divide and conquer principle by trying to break the problem into smaller dimensional ones like the hierarchical estimating of body pose in [55] (detect head first, then torso and so on) or the breaking of complex human movement into basic motions in [9].

There are also some opened related research areas that should be mentioned. First is the issue of human body pose estimation at multilevel (e.g. body level, head level, hand level) which was mentioned in [88]. We can see the benefits of having such a multilevel human body pose estimation system: Combined information from different level is more useful (e.g. in intelligent environment, the combination of body pose, hand pose, head pose would give better interpretation of human status/intention); Information from different levels can support each other and help to improve the estimation performance. However typical approaches in the area only deal with each task of body pose estimation, hand pose estimation, head pose estimation separately. Therefore, it is worth to have some studies that analyze the reasons why typical approaches only deal with one task at a time and find a way to achieve the goal of a full body model (e.g. including body, head, and hand). Another opened related research area that is worth to dealing with is the issue of pose estimation and tracking of multiple objects simultaneously.

Next, the sub-area of multi-view action recognition is reviewed, covering both 2D and 3D multi-view approaches, and publicly available multi-view datasets. A qualitative comparison of several promising approaches based on the IXMAS and i3DPost datasets, reveals that methods using 3D representations of the data turn out to outperform the 2D methods. Although the reviewed approaches show promising results for multi-view human body modeling, pose estimation and action recognition, 3D reconstructed data from multi-view camera systems has some shortcomings. First of all, the quality of the silhouettes is crucial for the outcome of applying Shape-from- Silhouettes. Hence, shadows, holes and other errors due to inaccurate foreground segmentation will affect the final quality of the reconstructed 3D data. Secondly, the number of views and the image resolution will influent the level of details which can be achieved, and self-occlusion is a known problem when reconstructing 3D data from multi-view image data, resulting in merging body parts. Finally, 3D data can only be reconstructed in a limited space where multiple camera views overlap.

In recent years other prominent vision-based sensors for acquiring 3D data have been developed. Time-of-Flight (ToF) range cameras, which are single sensors capable of measuring depth information, have become popular in the computer vision community. Especially, with the introduction of the Microsoft Kinect sensor [73], these single and direct 3D

imaging devices have become widespread and commercial available at low cost. Hence, the future of acquiring vision-based 3D data will move in this direction, and in the next years we will see many new proposed approaches for human body modeling, pose estimation, action recognition and other computer vision related topics.

Acknowledgements

This work has received funding from the Danish National Research Councils - FTP under the research project "Big Brother *is* watching you!". The authors thank the UC Discovery Digital Media Innovation (DiMI) program, NSF, and Volkswagen for their sponsorship. We would also like to thank our colleagues at CVRR-LISA laboratory, especially Dr. Shinko Cheng, Dr. Kohsia Huang, and Dr. Ivana Mikic for their valuable inputs.

References

[1] MuHAVi dataset instructions at http://dipersec.king. ac.uk/MuHAVi-MAS/.

[2] M. Ahmad and S.-W. Lee. Hmm-based human action recognition using multiview image sequences. In *ICPR*, 2006.

[3] A.A. Alatan, Y. Yemez, U. Güdükbay, X. Zabulis, K. Müller, C.E. Erdem, C. Weigel, and A. Smolic. Scene representation technologies for 3dtv - a survey. *EEE Trans. Circuits Syst. Video Techn.*, 17(11):1587–1605, 2007.

[4] A.A. Alonso, R.D Rosa, L.D. Val, M.I. Jimenez, and S. Franco. A robot controlled by blinking for ambient assisted living. *Proceedings of the 10th International Work-Conference on Artificial Neural Networks: Part II: Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, 2009.

[5] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. 3d shape histograms for similarity search and classification in spatial databases. In *SSD*, 1999.

[6] J. Assfalg, M. Bertini, C. Colombo, A.D. Bimbo, and W. Nunziati. Semantic annotation of soccer videos: automatic highlights identification. *CVIU*, 92(2-3), 2003.

[7] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, 2002.

[8] O. Bernier, P. Cheung-Mon-Chana, and A. Bougueta. Fast nonparametric belief propagation for real-time stereo articulated body tracking. *CVIU*, 113(1):29–47, 2009.

[9] F. Caillette, A. Galata, and T. Howard. Real-time 3-d human body tracking using learnt models of behaviour. *CVIU*, 109:112–125, 2008.

[10] C. Canton-Ferrer, J.R. Casas, and M. Pardás. Human model and motion based 3d action recognition in multiple view scenarios. In *EUSIPCO*, 2006.

[11] Y. Chen, L.B. Smith, S. Hongwei, A.F. Pereira, and T. Smith. Active information selection: Visual attention through the hands. *IEEE Transactions on Autonomous Mental Developmen*, 1(2):141–151, 2009.

[12] Shinko Y. Cheng and Mohan M. Trivedi. Multimodal voxelization and kinematically constrained gaussian mixture model for full hand pose estimation: An integrated systems approach. In *ICVS*, 2006.

[13] Shinko Y. Cheng and Mohan M. Trivedi. Articulated human body pose inference from voxel data using a kinematically constrained gaussian mixture model. In *CVPR Workshops*, 2007.

[14] S.Y. Cheng and M.M. Trivedi. Turn-intent analysis using body pose for intelligent driver assistance. *IEEE Pervasive Computing*, 5(4):28–37, 2006.

[15] S. Cherla, K. Kulkarni, A. Kale, and V. Ramasubramanian. Towards fast, view-invariant human action recognition. In *CVPR Workshops*, 2008.

[16] G. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematic estimation and motion capture. In *CVPR*, 2003.

[17] G. Cheung and T. Kanade. A real-time system for robust 3d voxel reconstruction of human motions. In *CVPR*, 2000.

[18] I. Cohen and H. Li. Inference of human postures by classification of 3d human body shape. In *AMFG*, 2003.

[19] S. Corazza, L. Mundermann, E. Gambaretto, G. Ferrigno, and T. Andriacchi. Markerless motion capture through visual hull, articulated icp and subject specific model generation. *IJCV*, 87(1-2), 2010.

[20] Q. Delamarre and O. Faugeras. 3d articulated models and multiview tracking with physical forces. *CVIU*, 81(3):328–357, 2001.

[21] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1-2), 2007.

[22] A. Farhadi and M.K. Tabrizi. Learning to recognize activities from the wrong view point. In *ECCV*, 2008.

[23] Preben Fihl and Thomas B. Moeslund. Invariant gait continuum based on the duty-factor. *SIViP*, 3(4):391–402, 2008.

[24] David Fofi, Tadeusz Sliwa, and Yvon Voisin. A comparative survey on invisible structured light. *Proc. SPIE*, 5303:90–98, 2004.

[25] J. Gall, B. Rosenhahn, T. Brox, and H. Seidel. Optimization and filtering for human motion capture: A multi-layer framework. *IJCV*, 87(1-2):75–92, 2010.

[26] J. Geigel and M. Schweppe. Motion capture for realtime control of virtual actors in live, distributed, theatrical performances. In *FG*, 2011.

[27] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, and I. Pitas. The i3dpost multi-view and 3d human action/interaction database. In *CVMP*, 2009.

[28] N. Gkalelis, N. Nikolaidis, and I. Pitas. View indepedent human movement recognition from multi-view video exploiting a circular invariant posture representation. In *ICME*, 2009.

[29] R. Gross and J. Shi. The cmu motion of body (mobo) database. In *Techical Report*, 2001.

[30] Anwaar Haq, Iqbal Gondal, and Manzur Murshed. On dynamic scene geometry for view-invariant action matching. In *CVPR*, 2011.

[31] M. Hofmann and D.M. Gavrila. Multi-view 3d human pose estimation in complex environment. *IJCV*, 2011.

[32] M.B. Holte, T.B. Moeslund, N. Nikolaidis, and I. Pitas. 3d human action recognition for multi-view camera systems. In *3DIMPVT*, 2011.

[33] K.S. Huang and M.M. Trivedi. 3d shape context based gesture analysis integrated with tracking using omni video array. In *CVPR Workshops*, 2005.

[34] P. Huang and A. Hilton. Shape-colour histograms for matching 3d video sequences. In *3DIM*, 2009.

[35] P. Huang, A. Hilton, and J. Starck. Shape similarity for 3d video sequences of people. *IJCV*, 89:362–381, 2010.

[36] Z. Husz and A. Wallace. Evaluation of a hierarchical partitioned particle filter with action primitives. In *CVPR Workshops*, 2007.

[37] B.-W. Hwang, S. Kim, and S.-W. Lee. A fullbody gesture database for automatic gesture recognition. In *FG*, 2006. http://gesturedb.korea.ac.kr/.

[38] A. Iosifidis, N. Nikolaidis, and I. Pitas. Movement recognition exploiting multi-view information. In *MMSP*, 2010.

[39] X. Ji and H. Liu. Advances in view-invariant human motion analysis: A review. *Trans. Sys. Man Cyber Part C*, 40(1):13–24, 2010.

[40] A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *PAMI*, 21(5):433–449, 1999.

[41] I.N. Junejo, E. Dexter, I. Laptev, and P. Pérez. Cross-view action recognition from temporal self-similarities. In *ECCV*, 2008.

[42] I.N. Junejo, E. Dexter, I. Laptev, and P. Pérez. View-independent action recognition from temporal self-similarities. *PAMI*, 33(1):172–185, 2011.

[43] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *SGP*, 2003.

[44] J. Kilner, J.-Y. Guillemaut, and A. Hilton. 3d action matching with key-pose detection. In *ICCV Workshops*, 2009.

[45] D. Knossow, R. Ronfard, and R. Horaud. Human motion tracking with a kinematic parameterization of extremal contours. *IJCV*, 79(3):247–269, 2008.

[46] A. Kolb, E. Barth, R. Koch, and R. Larsen. Time-of-flight sensors in computer graphics. In *Eurographics - State of the Art Reports*, 2009.

[47] M. Körtgen, M. Novotni, and R. Klein. 3d shape matching with 3d shape contexts. In *CESCG*, 2003.

[48] H. Lee and J. H. Kim. An hmm-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10), 1999.

[49] R. Li, T.P. Tian, S. Sclaroff, and M.H. Yang. 3d human motion tracking with a coordinated mixture of factor analyzers. *IJCV*, 87(1-2), 2010.

[50] J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. In *CVPR*, 2008.

[51] J. Liu and M. Shah. Learning human actions via information maximization. In *CVPR*, 2008.

[52] Jingen Liu, Mubarak Shah, Benjamin Kuipers, and Silvio Savarese. Cross-view action recognition via view knowledge transfer. In *CVPR*, 2011.

[53] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *CVPR*, 2007.

[54] Pyry Matikainen, Padmanabhan Pillai, Lily Mummert, Rahul Sukthankar, and Martial Hebert. Prop-free pointing detection in dynamic cluttered environments. In *FG*, 2011.

[55] Ivana Mikic, Mohan M. Trivedi, Edward Hunter, and Pamela Cosman. Human body model acquisition and tracking using voxel data. *IJCV*, 53(3):199–223, 2003.

[56] T.B. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *CVIU*, 81(3):231268, 2001.

[57] T.B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 104(2-3):90–126, 2006.

[58] L. Muendermann, S. Corazza, A.M. Chaudhari, T.P. Andriacchi, A. Sundaresan, and R. Chellappa. Measuring human movement for biomechanical applications using markerless motion capture. In *Proceeding of SPIE Three-Dimensional Image Capture and Applications*, 2006.

[59] E. Murphy-Chutorian and M.M. Trivedi. 3d tracking and dynamic analysis of human head movements and attentional targets. In *IEEE/ACM Int'l. Conf. on Distributed Smart Cameras*, 2008.

[60] E. Murphy-Chutorian and M.M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):607–626, 2009.

[61] E. Murphy-Chutorian and M.M. Trivedi. Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness. *IEEE Transactions on Intelligent Transportation Systems*, 2010.

[62] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Trans. Graph.*, 21:807–832, 2002.

[63] S. Park and M.M. Trivedi. Understanding Human Interactions with Track and Body Synergies (TBS) Captured from Multiple Views. *Computer Vision and Image Understanding*, 111(1):2–20, 2008.

[64] S. Pehlivan and P. Duygulu. A new pose-based representation for recognizing actions from multiple cameras. *CVIU*, 115:140–151, 2011.

[65] M. Pierobon, M. Marcon, A. Sarti, and S. Tubaro. 3-d body posture tracking for human action template matching. In *ICASSP*, 2006.

[66] Ronald Poppe. Evaluating example-based pose estimation: Experiments on the humaneva sets. In *CVPR Workshops*, 2007.

[67] Ronald Poppe. Vision-based human motion analysis: An overview. *CVIU*, 108(1-2):4–18, 2007.

[68] Ronald Poppe. A survey on vision-based human action recognition. *IVC*, 28(6):976–990, 2010.

[69] A.B. Postawa, M. Kleinsorge, J. Krueger, and G. Seliger. Automated image based recognition of manual work steps in the remanufacturing of alternators. *Advances in Sustainable Manufacturing*, 5:209–214, 2011.

[70] J. Radmer and J. Krueger. Depth data-based capture of human movement for biomechanical application in clinical rehabilitation use. In *5th International Symposium on Health Informatics and Bioinformatics*, 2010.

[71] K.K. Reddy, J. Liu, and M. Shah. Incremental action recognition using feature-tree. In *ICCV*, 2009.

[72] C. Rougier, J. Meunier, A. St-Arnaud, and J. Rousseau. Fall detection from human shape and motion history using video surveillance. In *21st International Conference on Advanced Information Networking and Applications Workshops*, 2007.

[73] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011.

[74] L. Sigal and M.J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. In *Techniacl Report*, 2006.

[75] L. Sigal and M.J. Black. Guest editorial: State of the art in image and video based human pose and motion estimation. *IJCV*, 87:1–3, 2010.

[76] G. Slabaugh, B. Culbertson, and T. Malzbender. A survey of methods for volumetric scene reconstruction for photographs. In *VG*, 2001.

[77] Yale Song, David Demirdjian, and Randall Davis. Multi-signal gesture recognition using temporal smoothing hidden conditional random fields. In *FG*, 2011.

[78] R. Souvenir and J. Babbs. Learning the viewpoint manifold for action recognition. In *CVPR*, 2008.

[79] J. Starck and A. Hilton. Surface capture for performance based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31, 2007.

[80] E. Stoykova, A.A. Alatan, P. Benzie, N. Grammalidis, S. Malasitis, J. Ostermann, S. Piekh, V. Sainov, C. Theobalt, T. Thevar, and X. Zabulis. 3-d time-varying scene capture technologies: A survey. *EEE Trans. Circuits Syst. Video Techn.*, 17(11):15681586, 2007.

[81] A. Sundaresan and R. Chellappa. Model driven segmentation of articulating humans in laplacian eigenspace. *PAMI*, 30(10):1771–1785, 2008.

[82] C. Tran, A. Doshi, and M.M. Trivedi. Pedal errors prediction by driver foot gesture analysis: A vision-based inquiry. In *IEEE Intelligent Vehicle Symposium*, 2011.

[83] C. Tran and M.M. Trivedi. Hand modeling and tracking from voxel data: An integrated framework with automatic initialization. In *IEEE International Conference on Pattern Recognition*, 2008.

[84] C. Tran and M.M. Trivedi. Human body modeling and tracking using volumetric representation: Selected recent studies and possibilities for extensions. In *ACM workshops*, 2008.

[85] C. Tran and M.M. Trivedi. Driver assistance for 'keeping hands on the wheel and eyes on the road. In *IEEE International Conference on Vehicular Electronics and Safety*, 2009.

[86] C. Tran and M.M. Trivedi. Introducing XMOB: Extremity Movement Observation Framework for Upper Body Pose Tracking in 3D. In *IEEE International Symposium on Multimedia*, 2009.

[87] D. Tran and A. Sorokin. Human activity recognition with metric learning. In *ECCV*, 2008.

[88] M.M. Trivedi. Human movement capture and analysis in intelligent environments. *Machine and Vision Applications*, 14(4):215–217, 2003.

[89] M.M. Trivedi and S.Y. Cheng. Holistic sensing and active displays for intelligent driver support systems. In *IEEE Computer Magazine*, 2007.

[90] M.M. Trivedi, S.Y. Cheng, E. Childers, and S. Krotosky. Occupant posture analysis with stereo and thermal infrared video: Algorithms and experimental evaluation. *IEEE Transactions on Vehicular Technology, Special Issue on In-Vehicle Vision Systems*, 53(6), 2004.

[91] M.M. Trivedi, K.S. Huang, and I. Mikic. Dynamic context capture and distributed video arrays for intelligent spaces. *IEEE Trans. on Systems, Man and Cybernetics, Part A*, 35(1):145–163, 2005.

[92] P. Turaga, A. Veeraraghavan, and R. Chellappa. Statistical analysis on stiefel and grassmann manifolds with applications in computer vision. In *CVPR*, 2008.

[93] A. Utasi and C. Benedek. A 3-d marked point process model for multi-view people detection. In *CVPR*, 2011.

[94] S.N. Vitaladevuni, V. Kellokumpu, and L.S. Davis. Action recognition using ballistic dynamics. In *CVPR*, 2008.

[95] A. Waibel, R. Stiefelhagen, R. Carlson, J. Casas, J. Kleindienst, L. Lamel, O. Lanz, D. Mostefa, M. Omologo, F. Pianesi, L. Polymenakos, G. Potamianos, J. Soldatos, G. Sutschet, and J. Terken. Computers in the human interaction loop. In *Handbook of Ambient Intelligence and Smart Environments, Springer*, 2010.

[96] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *CVIU*, 104(2):249–257, 2006.

[97] Daniel Weinland, Mustafa Özuysal, and Pascal Fua. Making action recognition robust to occlusions and viewpoint changes. In *ECCV*, 2010.

[98] Daniel Weinland, Rémi Ronfard, and Edmond Boyer. Action recognition from arbitrary views using 3d exemplars. In *ICCV*, 2007.

[99] Daniel Weinland, Rémi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *INRIA Report*, RR-7212:54–111, 2010.

[100] N. Werghi. Segmentation and modeling of full human body shape from 3-d scan data: A survey. *TSMC-C*, 37(6):1122–1136, 2007.

[101] P. Yan, S.M. Khan, and M. Shah. Learning 4d action feature models for arbitrary view action recognition. In *CVPR*, 2008.

[102] J. Ziegler, K. Nickel, and R. Stiefelhagen. Tracking of the articulated upper body on multi-view stereo image sequences. In *CVPR*, 2006.

Chapter 6

Multi-View Human Action Recognition

This chapter consists of the paper "A Local 3D Motion Descriptor for Multi-View Human Action Recognition from 4D Spatio-Temporal Interest Points" [A]. The paper presents a local feature descriptor-based strategy for 3D Human action recognition in multi-view video, where 3D motion descriptors are extracted locally from estimated 3D optical flow at detected 4D spatio-temporal interest points. The survey in chapter 5 reveals that such an approach has not yet been explored for 3D human action recognition. The paper is based on the spatio-temporal interest point detector and 3D optical flow described in chapter 2 and 3, respectively. Reference [B] describes intermediate work resulting in the final outcome in [A].

References

A. M.B. Holte, B. Chakraborty, J. Gonzàlez and T.B. Moeslund. A Local 3D Motion Descriptor for Multi-View Human Action Recognition from 4D Spatio-Temporal Interest Points. Submitted to *Journal of Selected Topics in Signal Processing, IEEE Signal Processing Society*, 2011.

B. M.B. Holte, T.B. Moeslund, N. Nikolaidis and I. Pitas. 3D Human Action Recognition for Multi-View Camera Systems. In *IEEE Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, Hangzhou, China*, May 2011.

A Local 3D Motion Descriptor for Multi-View Human Action Recognition from 4D Spatio-Temporal Interest Points

M.B. Holte, B. Chakraborty, J. Gonzàlez and T.B. Moeslund

Abstract

In this paper we address the problem of human action recognition in reconstructed 3-dimensional data acquired by multi-camera systems. We contribute to this field by introducing a novel 3D action recognition approach based on detection of 4D (3D space + time) Spatio-Temporal Interest Points (STIPs) and local description of 3D motion features. STIPs are detected in multi-view images and extended to 4D using 3D reconstructions of the actors and pixel-to-vertex correspondences of the multi-camera setup. Local 3D motion descriptors, Histogram of Optical 3D Flow (HOF3D), are extracted from estimated 3D optical flow in the neighborhood of each 4D STIP and made view-invariant. The local HOF3D descriptors are divided using 3D spatial pyramids to capture and improve the discrimination between arm- and leg-based actions. Based on these pyramids of HOF3D descriptors we build a Bag-of-Words (BoW) vocabulary of human actions, which is compressed and classified using Agglomerative Information Bottleneck (AIB) and Support Vector Machines (SVM), respectively. Experiments on the publicly available i3DPost and IXMAS datasets show promising state-of-the-art results and validate the performance and view-invariance of the approach.

6.1 Introduction

Using multi-camera setups for human action recognition has gained tremendous attention in recent years, due to its large application area, e.g., Human-Computer Interaction (HCI), intelligent environment, augmented reality, 3D gaming, local surveillance, mobile devices etc. Several interesting approaches in the field of 3D human action recognition exist in literature [44, 49, 65, 22], which explore 3D representation of the acquired multi-view data for robust action recognition.

A 3D data representation is more informative than the analysis of 2D activities carried out in the image plane, which is only a projection of the actual actions. As a result, the projection of the actions will depend on the viewpoint, and not contain full information about the performed activities. To overcome this shortcoming the use of 3D data has been introduced through the use of two or more cameras. [8, 12, 53, 62]. In this way the surface structure or a 3D volume of the person can be reconstructed, e.g., by Shape-From-Silhouette (SFS) techniques [56], and thereby a more descriptive representation for action recognition can be established.

2D human action recognition has moved from model-based approaches to model-free approaches using local motion features. In this context, methods based on Spatio-Temporal Interest Points (STIPs) and Bag-of-Words (BoW) are successfully applied to this area. On the contrary, 3D Human action recognition is more confined towards model-based approaches or holistic features. To minimize this gap, we contribute to the field of multi-view human action recognition, by introducing a novel 3D action recognition approach based on detection of 4D Spatio-Temporal Interest Points and local description of 3D motion features extracted from reconstructed 3D data acquired by multi-camera systems. Opposed to other methods for 3D action recognition, which are solely based on holistic features, e.g. [16, 47, 55, 62], our approach extends the concepts of STIP detection and local feature description for building a Bag-of-Words (BoW) vocabulary of human actions, which has gained popularity in the 2D image domain, to the 3D case.

6.1.1 Related Work

The use of 3D data allows for efficient analysis of 3D human activities. However, we are still faced with the problem that the orientation of the subject in the 3D space should be known. Therefore, approaches have been proposed without this assumption by introducing view-invariant or view-independent representations.

View-Invariant 2D Feature Description

One line of work concentrates solely on the 2D image data acquired by multiple cameras [13, 20, 22, 55]. In the work of Souvenir et al. [55] actions are described in a view-invariant manner by computing $\mathcal{R}$ transform surfaces of silhouettes and manifold learning. Gkalelis et al. [13] exploit the circular shift invariance property of the discrete Fourier Transform (DFT) magnitudes, and use Fuzzy Vector Quantization (FVQ) and Linear Discriminant Analysis (LDA) to represent and classify actions. Another approach

was proposed by Iosifidis et al. [20], where binary body masks from frames of a multi-camera setup are concatenated to multi-view binary masks.

Some authors perform action recognition from image sequences in different viewing angles. Ahmad et al. [1] apply Principal Component Analysis (PCA) of optical flow velocity and human body shape information, and then represent each action using a set of multi-dimensional discrete Hidden Markov Models (HMM) for each action and viewpoint. Cherla et al. [7] show how view-invariant recognition can be performed by using data fusion of two orthogonal views. An action basis is built using eigenanalysis of walking sequences of different people, and projections of the width profile of the actor and spatio-temporal features are applied. Finally, Dynamic Time Warping (DTW) is used for recognition. A number of other techniques have been employed, like metric learning [58] or representing action by feature-trees [50] or ballistic dynamics [61]. In [63] Weinland et al. propose an approach which is robust to occlusions and viewpoint changes using local partitioning and hierarchical classification of 3D Histogram of Oriented Gradients (3DHOG) volumes.

Others use synthetic data rendered from a wide range of viewpoints to train their model and then classify actions in a single view, e.g. Lv et al. [43], where shape context is applied to represent key poses from silhouettes and Viterbi Path Searching for classification. A similar approach was proposed by Fihl. et al. [11] for gait analysis.

Another topic which has been explored by several authors the last couple of years is cross-view action recognition. This is a difficult task of recognizing actions by training on one view and testing on another completely different view (e.g., the side view versus the top view of a person in IXMAS). A number of techniques have been proposed, stretching from applying multiple features [37], information maximization [39], dynamic scene geometry [14], self similarities [24, 25] and transfer learning [10, 40]. For additional related work on view-invariant approaches please refer to the recent survey by Ji et al. [22].

3D Shape and Pose Descriptors

Another line of work utilize the full reconstructed 3D data for feature extraction and description. ([2, 23, 27, 30, 46]). Johnson and Hebert proposed the spin image [23], and Osada et al. the shape distribution [46]. Ankerst et al. introduced the shape histogram [2], which is a similar to the 3D extended shape context [3] presented by Körtgen et al. [30], and Kazhdan et al. applied spherical harmonics to represent the shape histogram in a view-invariant manner [27]. Later Huang et al. extended the shape histogram with color information [17]. Recently, Huang et al. made a comparison of these shape descriptors combined with self similarities, with the shape histogram (3D shape context) as the top performing descriptor. [18, 19].

A common characteristic of all these approaches is that they are solely based on static features, like shape and pose description, while the most popular and best performing 2D image descriptors apply motion information or a combination of the two [44, 34, 38, 65].

3D Motion Descriptors

Instead of only relying on static features, some authors add temporal information by capturing the evolvement of static descriptors over time, i.e., shape and pose changes, by accumulating static descriptors over time, track human shape or pose information, or apply sliding windows [47, 48, 62, 64]. Cohen et al. [8] use 3D human body shapes for view-invariant identification of human body postures, which later was used by Pierobon et al. [48] for human action recognition. The Motion History Volume (MVH) was proposed by Weinland et al. [62], as a 3D extension of Motion History Images (MHIs) [4]. MHVs are created by accumulating static human postures over time in a cylindrical representation, which is made view-invariant with respect to the vertical axis by applying the Fourier transform in cylindrical coordinates. The same representation was used by Turaga et al. [59] in combination with a more sophisticated action learning and classification based on Stiefel and Grassmann manifolds. Later, Weinland et al. [64] proposed a framework, where actions are modeled using 3D occupancy grids, built from multiple viewpoints, in an exemplar-based Hidden Markov Models (HMM). Learned 3D exemplars are used to produce 2D image information which is compared to the observations, hence, 3D reconstruction is not required during the recognition phase.

Pehlivan et al. [47] present a view-independent representation based on human poses. The volume of the human body is first divided into a sequence of horizontal layers, then circular features in all layers are used to generate pose descriptors in an action sequence, which are combined to generate motion descriptors. Action recognition is then performed with a simple nearest neighbor classifier. A different strategy is presented by Yan et al. [68]. They propose a 4D action feature model (4D-AFM) for recognizing actions from arbitrary views based on spatio-temporal features of spatio-temporal volumes (STVs) [69]. The extracted features are mapped from the STVs to a sequence of reconstructed 3D visual hulls over time, resulting in the 4D-AFM model, which is used for matching actions. Another pair of 3D descriptors which are based on rich motion information are the 3D Motion Context (3D-MC) and the Harmonic Motion Context (HMC) proposed by Holte et al. [16] The 3D-MC descriptor is a motion oriented 3D version of the shape context [3, 30], which incorporates motion information implicitly from 3D optical flow. The HMC descriptor is an extended version of the 3D-MC descriptor that makes it view-invariant by decomposing the representation into a set of spherical harmonic basis functions.

Spatio-Temporal Interest Points

In common for these approaches is that they are all based on holistic feature representation of the human body and its motion. In contrast, recent progress in the field of video-based 2D human action recognition points towards the use of Spatio-Temporal Interest Points (STIPs) for local descriptor-based recognition strategies. Laptev and Lindeberg first proposed STIPs for action recognition [31], by introducing a space-time extension of the popular Harris detector [15]. They detect regions having high intensity variation in both space and time as spatio-temporal corners. It usually suffers from sparse STIP detection. Later other methods for detecting STIPs have been reported. [9, 21, 45, 66, 67]. Dollar et al. [9] improved the sparse STIP detector by applying temporal Gabor filters and select regions of high responses. Dense and scale-invariant spatio-temporal

interest points were proposed by Willems et al. [66], as a spatio-temporal extension of the Hessian saliency measure, previously applied for object detection. Instead of applying local information for STIP detection Wong et al. [67] propose a global information-based approach. They use global structural information of moving points and select STIPs according to their probability of belonging to the relevant motion. Recently, Chakraborty et al. [5] designed a selective STIP detector for recognition of human actions, which splits up the spatial and temporal computation in two steps. First, it incorporates surround suppression of the output of the basic Harris corner detector [15]. Hereafter, local spatio-temporal constraints are imposed to obtain a final set of STIPs which is more robust, while suppressing unwanted background STIPs.

Local Image Descriptors

For describing the local image region properties in the neighborhoods of the detected STIPs, several local descriptors have been proposed in the past few years [9, 66, 28, 29, 33, 34, 51]. Local feature descriptors extract shape and motion information using image measurements, such as spatial or spatio-temporal image gradients or optical flow. Laptev et al. [34] introduced a combined descriptor to characterize local motion and appearance by computing Histograms of Spatial Gradients (HOG) and Optic Flow (HOF) accumulated in space-time neighborhoods of detected interest points. Willems et al. [66] proposed the Extended SURF (ESURF) descriptor, which extends the image SURF descriptor to videos. The authors divide 3D patches into cells, where each cell is represented by a vector of weighted sums of uniformly sampled responses of the Haar-wavelets along the three axes. Dollar et al. [9] proposed the *Cuboid* descriptor along with their detector. The authors concatenate the gradients computed for each pixel in the neighborhood into a single vector and apply Principal Component Analysis (PCA) to project the feature vector onto a low dimensional space. Compared to the HOG-HOF descriptor proposed by Laptev et al. [34], it does not distinguish the appearance and motion features. The 3D-SIFT descriptor was developed by Scovanner et al. [51]. This descriptor is similar to the Scale Invariant Feature Transformation (SIFT) descriptor [41], except that it is extended to video sequences by computing the gradient direction for each pixel spatio-temporally in three-dimensions. Another extension of the popular SIFT descriptor was proposed by Kläser et al. [28]. It is based on histograms of 3D gradient orientations, where gradients are computed using an integral video representation. Finally, a prominant descriptor is the N-jets. [29, 32]. An N-jet is the set of partial derivatives of a function up to order N, and is usually computed from a scale-space representation.

Although STIP detection and local motion feature descriptors have proven to be very successful for video-based 2D human action recognition, the concept has yet to be applied to the 3D domain of action recognition, where model-based techniques or holistic features are still dominating. Li et al. [35] proposed an approach based on bag of 3D points, randomly sampled at the silhouette/contour of the human body in depth images. However, the sampled contour points only describe randomly extracted static information. In contrast, STIPs are detected at positions with significant and descriptive motion regions, and a feature descriptor like HOF is based on motion information, where optical flow is always giving a true measurement of the motion.

6.1.2 Our Approach and Contributions

In this work we perform 3D human action recognition using video data acquired by multi-view camera systems and reconstructed 3D models. The contributions of this paper are as follows: (1) We propose a novel 3D action recognition approach based on detection of 4D (3D space + time) STIPs and local description of 3D motion features. STIPs are detected in multi-view images in a selective manner by surround suppression of the output of the basic Harris corner detector and imposing local spatio-temporal constraints [5]. Hereafter, the multi-view image STIPs are extended to 4D using 3D reconstructions of the actors and pixel-to-vertex correspondences of the multi-camera setup (section 6.2). (2) By introducing a novel local 3D motion descriptor, called Histogram of Optical 3D Flow (HOF3D), we represent estimated 3D optical flow [16] in the neighborhood of each 4D STIP, and examine four solutions to make the HOF3D descriptor view-invariant (section 6.3): (i) vertical rotation with respect to the orientation of the normal vector and (ii) the orientation of the velocity vector, (ii) circular bin shifting with respect to the horizontal mode of the histogram and (iv) by decomposing the representation into a set of spherical harmonic basis functions. (3) The local HOF3D descriptors are divided using 3D spatial pyramids to capture and improve the discrimination between arm- and leg-based actions. In section 6.4 we examine two pyramid divisions based on a horizontal plane estimated as (i) the center of gravity of the 3D human model and (ii) the center of gravity of the detected STIPs. Based on these pyramids of HOF3D descriptors we build a Bag-of-Words (BoW) vocabulary of human actions, which is compressed and classified using Agglomerative Information Bottleneck (AIB) and Support Vector Machines (SVM), respectively. (4) Experiments reported in section 7.5 on the publicly available i3DPost and IXMAS datasets show promising state-of-the-art results and validate the performance and view-invariance of the approach. Finally, in section 6.6 we give some concluding remarks.

6.2 4D Spatio-Temporal Interest Point Detection

We detect STIPs using the selective STIP detector proposed by [5], which first detects spatial interest points (SIPs), then perform surround suppression, impose local spatio-temporal constraints and scale adaption, to obtain a final set of STIPs. Hereafter, we extend the detected STIPs to 4D STIPs using pixel-to-vertex correspondences (Fig. 6.1).

6.2.1 Selective STIPs

The detector applies the basic Harris corner detector [15] and computes the first set of interest points:

$$C_\sigma(x,y) = \frac{I_x^2 I_y^2 - I_{xy}^2}{I_x^2 + I_y^2 + \epsilon} \tag{6.1}$$

where σ is the spatial scale; I_x, I_y and I_{xy} are the partial derivatives over x, y and xy, respectively; and ϵ is a small constant. Apart from the detected SIPs on the human actors, the spatial corners C_σ contain a significant amount of unwanted background SIPs [5].

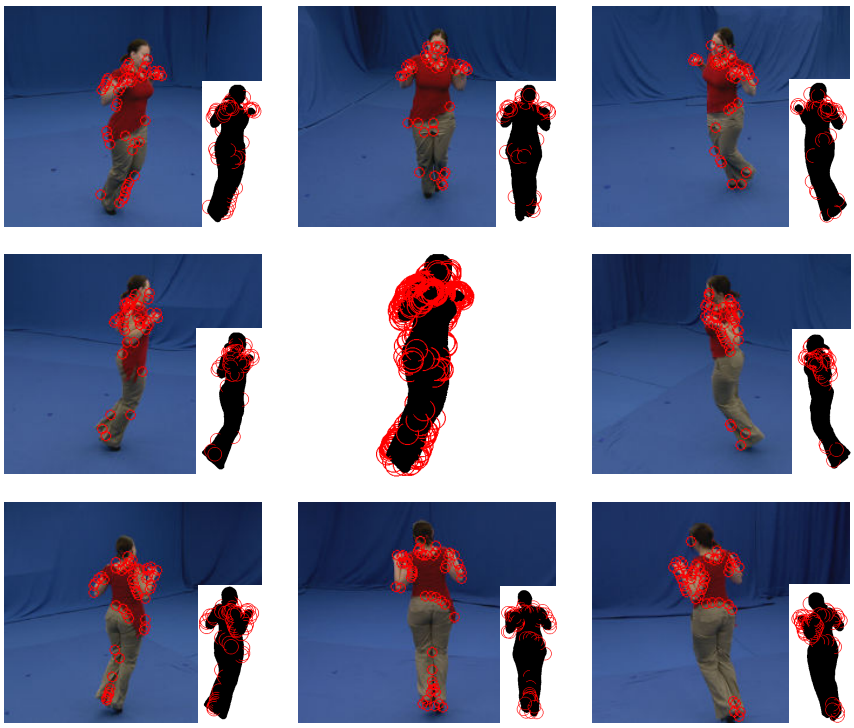


Figure 6.1: Detection of STIPs in multi-frames, and extension to 4D STIPs using 3D reconstructions of the actors and pixel-to-vertex correspondences, for extraction of local 3D motion descriptors.

Surround Suppression

A surround suppression mask (SSM) at each interest point is employed, taking the current point under evaluation as the centre of the mask, in order to eliminate these unwanted background SIPs. The influence of all surrounding points of the mask on the central point is determined, and accordingly a suppression decision is taken. Surround suppression is implemented by computing an inhibition term for each point of C_σ. For this purpose a gradient weighting factor $\triangle_{\Theta,\sigma}(\mathbf{X}, \mathbf{X}_{\mathbf{u},\mathbf{v}})$ is introduced, which is defined:

$$\triangle_{\Theta,\sigma}(\mathbf{X}, \mathbf{X}_{\mathbf{u},\mathbf{v}}) = |\cos(\Theta_\sigma(\mathbf{X}) - \Theta_\sigma(\mathbf{X}_{\mathbf{u},\mathbf{v}})| \tag{6.2}$$

where $\Theta_\sigma(\mathbf{X})$ and $\Theta_\sigma(\mathbf{X}_{\mathbf{u},\mathbf{v}})$ are the gradients at point $\mathbf{X} \equiv (x, y)$ and $\mathbf{X}_{\mathbf{u},\mathbf{v}} \equiv (x-u, y-v)$, respectively; u and v define the horizontal and vertical range of the SSM. If $\Theta_\sigma(\mathbf{X})$ and $\Theta_\sigma(\mathbf{X}_{\mathbf{u},\mathbf{v}})$ are identical, the weighting factor attains its maximum ($\triangle_{\Theta,\sigma} = 1$), while the value of the factor decreases with the angle difference and reaches a minimum ($\triangle_{\Theta,\sigma} = 0$), when the two gradient orientations are orthogonal. Hence, the surrounding interest points which have the same orientation, as that of $\mathbf{X}$, will have a maximal inhibitory effect.

For each interest point $C_\sigma(\mathbf{X})$, a suppression term $t_\sigma(\mathbf{X})$ is defined as the weighted sum of gradient values in the suppression surround of that point:

$$t_\sigma(\mathbf{X}) = \int\int_\Omega C_\sigma(\mathbf{X}_{\mathbf{u},\mathbf{v}}) \times \triangle_{\Theta,\sigma}(\mathbf{X}, \mathbf{X}_{\mathbf{u},\mathbf{v}}) du dv \tag{6.3}$$

where Ω is the image coordinate domain. An operator $C_{\alpha,\sigma}(\mathbf{X})$ is introduced, which takes

its inputs: the corner magnitude $C_\sigma(\mathbf{X})$ and the suppression term $t_\sigma(\mathbf{X})$:

$$C_{\alpha,\sigma}(\mathbf{X}) = H(C_\sigma(\mathbf{X}) - \alpha \times t_\sigma(\mathbf{X})) \tag{6.4}$$

where $H(z) = z$ when $z \geq 0$ and *zero* for negative z values, and α controls the strength of the surround suppression.

Local Spatio-Temporal Constraints

Local spatio-temporal constraints are imposed by non-maxima suppression of the surround suppression responses $C_{\alpha,\sigma}$ (Equation 6.4), and scale adaption is achieved by applying a multi-scale approach [34] and compute suppressed STIPs in *five* different scales $S_\sigma = \{\frac{\sigma}{4}, \frac{\sigma}{2}, \sigma, 2\sigma, 4\sigma\}$. We follow the idea of scale selection presented by Lindeberg [36] to keep the best set of STIPs obtained for each scale. The best scales are selected by maximizing the normalized differential invariant,

$$\tilde{\kappa}_{norm} = \sigma_0^{2\gamma} L_y L_{xx} \tag{6.5}$$

where $L = g(\cdot; \sigma_0, \tau_0) \otimes I$, i.e. the image I is convoluted with the Gaussian kernel g; L_y is the first order y derivative and L_{xx} is the second order x derivative of L. Lindeberg [36] reports that $\gamma = \frac{7}{8}$ performs well in practice to achieve the maximum value of $(\tilde{\kappa}_{norm})^2$ for spatial interest point detected at multiple scales.

For the temporal constraints, a frame-wise interest point matching algorithm is applied [26], and the points are kept based on the 1D Gabor filter response in the temporal direction of the matching spatial interest points.

6.2.2 4-Dimensional STIPs

After detection of STIPs in multi-frame images we extend the resulting interest points into 4D STIPs. For this purpose we use the camera calibration data for the multi-view camera system [12], and project the vertices $\mathbf{p}$ of reconstructed 3D mesh models [56] onto the respective image planes with coordinates (u, v), using the following set of equations:

$$\begin{aligned} \mathbf{p}_c &= R_i \mathbf{p} + t_i \\ r &= \sqrt{d_x^2 + d_y^2},\ d_x = f_{i,x}\frac{p_{c,x}}{p_{c,z}}, d_y = f_{i,y}\frac{p_{c,y}}{p_{c,z}} \\ (u, v) &= \big(c_{i,x} + d_x(1 + k_{i,1}r), c_{i,y} + d_y(1 + k_{i,1}r)\big) \end{aligned} \tag{6.6}$$

where R and t are the camera rotation matrix and translation vector; f_x and f_y are the x and y components of the focal length f; c_x and c_y are the x and y components of the principal point c, and k_1 is the coefficient of a first order distortion model for the i^{th} camera, respectively. Since multiple vertices might be projected onto the same image pixel, we create a z-buffer containing the depth ordered vertices $\mathbf{p}_d$, and select the vertex with the shortest distance to the respective camera. The distance d is determined with respect to the centre of projection $\mathbf{o}$, as follows:

$$\begin{aligned} \text{z-buffer} &= [\mathbf{p}_{d,1}, \mathbf{p}_{d,2}, \ldots, \mathbf{p}_{d,n}] \\ d &= |\mathbf{p}_i - \mathbf{o}_i|, \text{ where } \mathbf{o}_i = -R_i^T t_i \end{aligned} \tag{6.7}$$

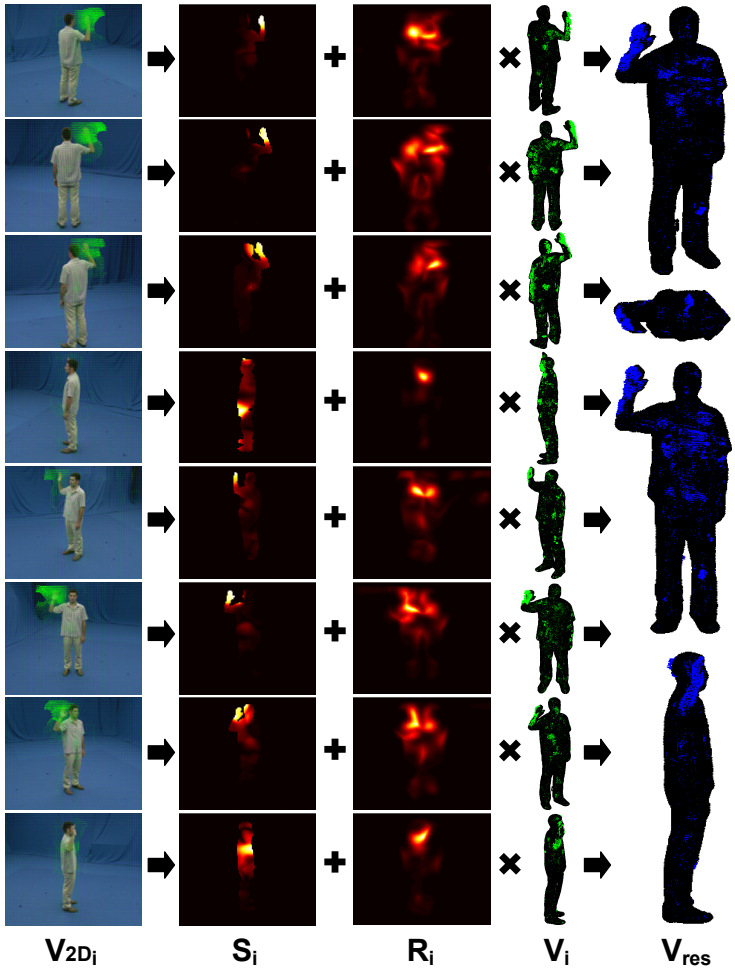
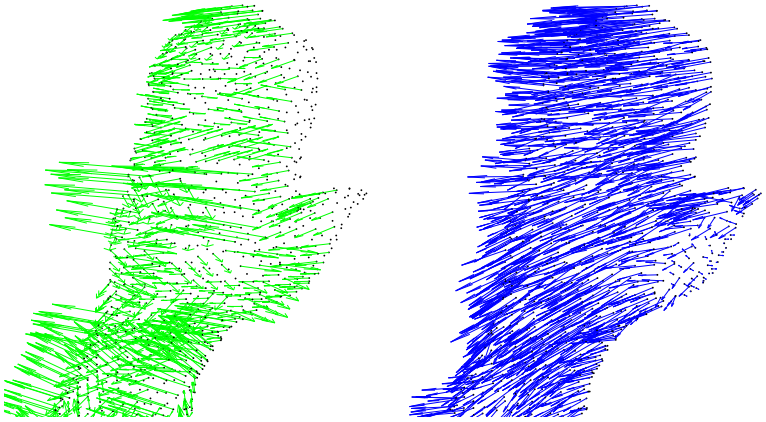


Figure 6.2: A schematic overview of the computation of 3D optical flow $\mathbf{V}_{\text{res}}$, by fusing optical flow estimated in multi-frames $\mathbf{V}_{2D,i}$, extended to 3D flow $\mathbf{V}_i$, and weighted by the significance of local motion $\mathbf{S}_i$ and it reliability $\mathbf{R}_i$.

This has proven to work well for selecting the best corresponding vertices in case of multiple instances [16]. Figure 6.1 present an example of 4D STIP detection.

6.3 Local 3D Motion Description

We detect motion in Multi-frames $\mathcal{F} = (I_1, I_2, \ldots, I_n)$, which is a set of image frames I acquired by n synchronized cameras, using a 3D version of optical flow [16] to produce *velocity annotated point clouds* [57] or *scene flow* [60] (3D optical flow), and combine the estimated 3D optical flow for each view (Fig. 6.2, 6.3 and 6.4). The estimated 3D optical flow is represented efficiently by introducing a local 3D motion descriptor, Histogram of 3D Optical Flow (HOF3D), which is made view-invariant.



(a) $\mathbf{V}_i$ (b) $\mathbf{V}_{\text{res}}$

Figure 6.3: Examples of (a) single-view 3D optical flow and (b) combined 3D optical flow.

6.3.1 3-Dimensional Optical Flow

Optical flow is computed using the Lucas and Kanade algorithm [42] for each multi-frame $\mathcal{F}_i$ of a multi-view sequence of images $(\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_m)$, and based on data from two consecutive multi-frames $(\mathcal{F}_i, \mathcal{F}_{i-1})$. Each pixel of multi-frame $\mathcal{F}_i$ is annotated with a 2D velocity vector $\mathbf{v}_{2D} = (v_x, v_y)^T$ (see Figure 6.2), resulting in temporal pixel correspondences between multi-frame $\mathcal{F}_i$ and $\mathcal{F}_{i-1}$.

For each pixel in the multi-frames we transform the temporal pixel correspondences into temporal 3D vertex correspondences $(\mathbf{p}_k^i, \mathbf{p}_l^{i-1})$ (Equation 6.6 and 6.7), which can be used to compute 3D velocities $\mathbf{v}_{3D} = (v_x, v_y, v_z)^T = \mathbf{p}_k^i - \mathbf{p}_l^{i-1}$. Figure 6.2 and 6.3.a present examples of estimated 3D optical flow. The 3D optical flow for each view $\mathbf{V}_i$ is combined into a resulting 3D optical flow $\mathbf{V}_{\text{res}}$, by weighting each component by the significance $\mathbf{S}_i$ of local motion and the reliability $\mathbf{R}_i$ of the estimated optical flow, as given by Equation 6.8:

$$\mathbf{V}_{\text{res}} = \sum_{i=1}^{n} \left(\alpha \frac{\mathbf{S}_i}{\sum_{k=1}^{n} \mathbf{S}_k} + \beta \frac{\mathbf{R}_i}{\sum_{l=1}^{n} \mathbf{R}_l} \right) \mathbf{V}_i \tag{6.8}$$

where n is the number of camera views, α and β are weights of the two measurements, such that $\alpha + \beta = 1$ (we set $\alpha = 0.75$ and $\beta = 0.25$). Since we focus on motion vectors, we are interested in robust and significant motion. Therefore, we apply a weight $\mathbf{S} = \sqrt{v_{2D,x}^2 + v_{2D,y}^2}$ to each of the velocity components (v_x, v_y, v_z) falling within the region of interest, determined by the projected silhouettes of the 3D models onto the respective image planes. In this way we give emphasis to the velocity components based on the total length of the 2D optical flow vector, i.e., the significance of local motions. This had proven to be an important asset, reducing the impact of erroneous 3D motion vectors, when falsified pixel-to-vertex correspondences have been established. The reliability $\mathbf{R}$ is a measure of the "cornerness" of the gradients in the window used to estimate optical flow, and is determined by the smallest eigenvalue $\mathbf{R} = \lambda_2$ of the second moment matrix. In this way we check for ill conditioned second moment matrices, and give emphasis to

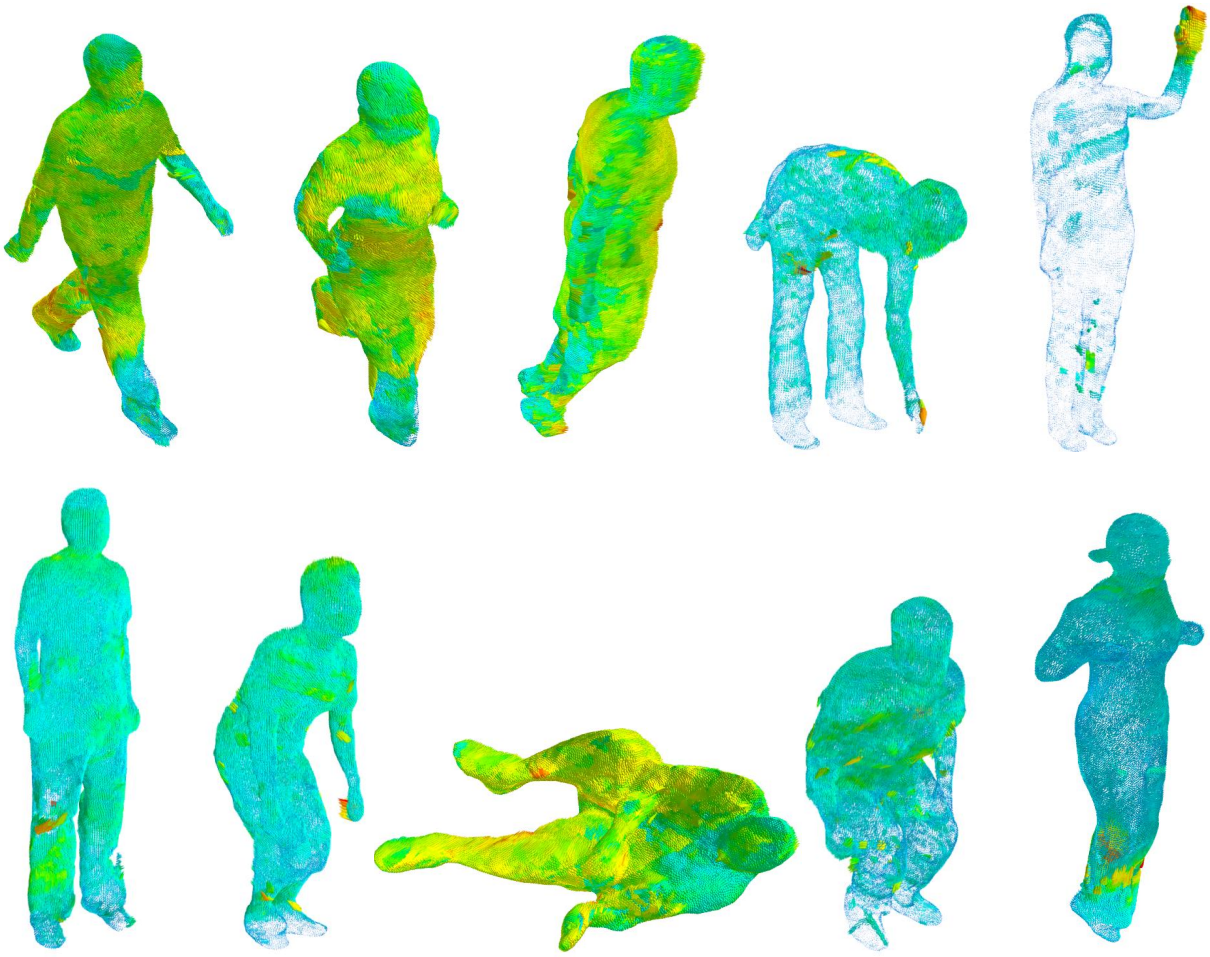


Figure 6.4: Examples of the resulting 3D optical flow for the 10 actions performed by 8 actors in the i3DPost dataset. The velocity vectors are color coded, so blue corresponds to low velocities and yellow/red corresponds to high velocities.

flow components based on their reliability. Figure 6.2, 6.3.b and 6.4 show examples of the resulting 3D optical flow.

6.3.2 Histogram of 3D Optical Flow

The extracted 3D motion in the form of 3D optical flow is represented efficiently by introducing a local 3D motion descriptor, Histogram of 3D Optical Flow (HOF3D), which is based on similar concepts as the HOF image descriptor proposed by Laptev et al. [34]. It is based on a spherical histogram, which is centered in the detected STIP and divided linearly into S azimuthal (east-west) orientation bins and T colatitudinal (north-south) bins (see Figure 6.5). For each bin of the histogram the velocity vector of each vertex falling within that particular bin, within a spherical support region with radius r, is accumulated and weighted by the length of the velocity vector. Hence, the descriptor captures both the location of motion, together with the amount of motion and its direction. We set $S = 8$, $T = 4$ and $r = 100$ mm, resulting in a $S \times T = 32$ dimensional feature vector for each STIP.

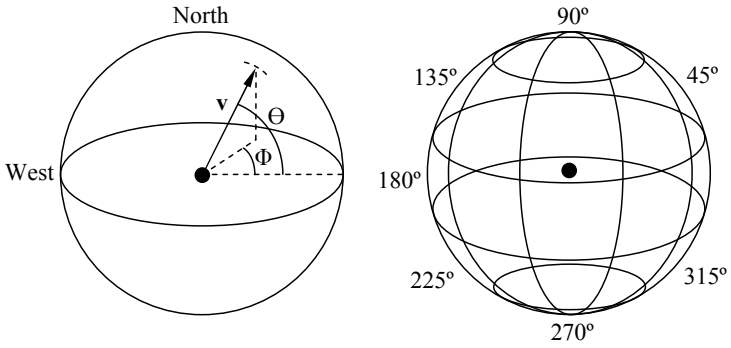


Figure 6.5: The HOF3D descriptor and its subdivision into 8 azimuthal and 4 colatitudinal bins.

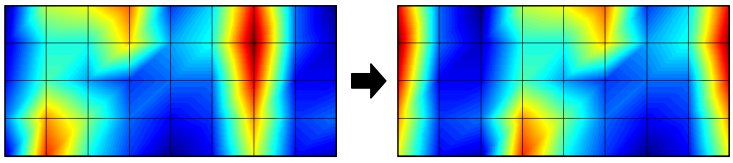


Figure 6.6: Circular bin shifting of the HOF3D histogram with respect to the horizontal mode of the histogram ($\text{HOF3D}_{\text{mode}}$).

In the Scale Invariant Feature Transform (SIFT) [41], partial invariance to the effect of illumination changes on the gradient magnitude is imposed by thresholding and normalizing the feature vector. In the same way we impose partially invariance to the velocity of movements, like in the case where two individuals perform the same action at different speed. Hence, the feature vector gives greater emphasis to the location and orientation, while reducing the influence of large velocity values.

6.3.3 View-Invariance

View-invariance is an essential criterion of feature description and recognition in 3D, since a feature (in our case the direction of extracted motion) might appear very differently depending on the viewpoint. For view-invariant human action recognition it is sufficient to consider the variations around the vertical axis of the human body. In the following we propose four solutions to transform the HOF3D descriptors into view-invariant representations: (i) vertical rotation with respect to the orientation of the normal vector and (ii) the orientation of the velocity vector, (ii) circular bin shifting with respect to the horizontal mode of the histogram, and (iv) by decomposing the representation into a set of spherical harmonic basis functions.

Vertical Rotation

The HOF3D descriptor is rotated around the vertical axis with respect to an azimuthal reference orientation $\angle\theta_{ref}$ of the evaluated STIP: $\angle\theta - \angle\theta_{ref}$. We evaluate two reference orientations. The orientation of the 3D models normal vector ($\text{HOF3D}_{\text{norm}}$) and the orientation of the velocity vector of the 3D optical flow ($\text{HOF3D}_{\text{flow}}$) at that particular

STIP.

Circular Bin Shifting

We perform circular bin shifting of the histogram with respect to the horizontal mode of the histogram ($\text{HOF3D}_{\text{mode}}$). The horizontal mode is determined as the set of vertical orientation bins with the largest value. An example is given in Figure 6.6.

Spherical Harmonics

Finally, the HOF3D descriptor is made view-invariant with respect to the vertical axis by decomposing the spherical Histogram representation $f(\theta, \phi)$ into a weighted sum of spherical harmonics (HHOF3D), as given by Equation 6.9.

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} A_l^m \, Y_l^m(\theta, \phi) \tag{6.9}$$

where the term A_l^m is the weighing coefficient of *degree* m and *order* l, while the complex functions $Y_l^m(\cdot)$ are the actual spherical harmonic functions of *degree* m and *order* l. The complex function $Y_l^m(\cdot)$ is given by Equation 6.10.

$$Y_l^m(\theta, \phi) = K_l^m \, P_l^{|m|}(\cos\theta) \, e^{jm\phi} \tag{6.10}$$

The term K_l^m is a normalization constant, while the function $P_l^{|m|}(\cdot)$ is the *associated Legendre Polynomial*. The key feature to note from Equation 6.10 is the encoding of the azimuthal variable ϕ, which solely inflects the *phase* of the spherical harmonic function and has no effect on the *magnitude*. This effectively means that $||A_l^m||$, i.e. the norm of the decomposition coefficients of Equation 6.9 is invariant to parameterization in the variable ϕ.

The actual determination of the spherical harmonic coefficients is based on an inverse summation as given by Equation 6.11, where N is the number of samples $(S \times T)$, and $4\pi/N$ is the surface area of each sample on the unit sphere.

$$(A_l^m)_f = \frac{4\pi}{N} \sum_{\phi=0}^{2\pi} \sum_{\theta=0}^{\pi} f(\theta, \phi) \; Y_l^m(\theta, \phi) \tag{6.11}$$

In a practical application it is not necessary (or possible, as there are infinitely many) to keep all coefficient A_l^m. Contrary, it is assumed the functions f are band-limited, hence it is only necessary to keep coefficient up to some bandwidth $l = B$, where the dimensionality becomes $D = (B+1)(B+2)/2$. Concretely, we set $B = 15$, resulting in 136 coefficients.

6.4 Vocabulary Building and Classification

We apply a BoW model to learn the visual vocabularies of the extracted HOF3D descriptors. We extend the idea of [38] by introducing pyramid levels in the feature space, but

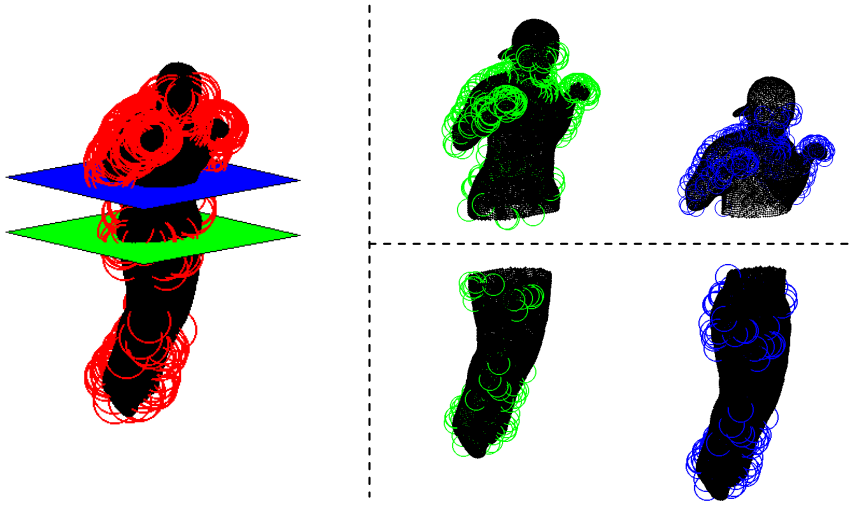


Figure 6.7: 3D spatial pyramid of level 2 with division by a horizontal plane estimated by the center of mass of the reconstructed model (a) and the detected 4D STIPs (b).

instead of applying a pyramid at feature level, as in [39], we apply it at STIP level in a 3D coordinate system. This makes the problem of grouping the local features much simpler yet robust, since our STIPs are detected in a selective and robust manner. Finally, we apply vocabulary compression, at each pyramid level, to reduce the dimensionality of the feature space.

6.4.1 3D Spatial Pyramids

Let I_T be the T^{th} frame of the image sequence I. We then quantize this the set of detected STIPs into q levels, $\mathcal{S} = \{s_0, s_1, \ldots, s_{q-1}\}$. We examine two solutions for pyramid divisions based on a horizontal plane estimated as (i) the center of gravity of the 3D human model (SP_{model}) and (ii) the center of gravity of the detected STIPs (SP_{STIPs}). Accordingly, we group the HOF3D descriptors into different levels of the pyramid. The structure of the 2-level 3D spatial pyramid is illustrated in Figure 6.7. This horizontal division helps to capture the distinguishing characteristics of arm- and leg-based actions. We do not apply further pyramid levels or vertical division, since this will conflict with the view-invariance of the approach.

6.4.2 Vocabulary Compression

After dividing the HOF3D descriptors into the described pyramid levels, we create initial vocabularies of a relatively large size (200 words). To reduce the final dimensionality of the feature space, we use vocabulary compression, as in [38], but at each level of the pyramid to achieve a compact yet discriminative visual-word representation of actions.

Let A be a discrete random variable which takes the value of a set of action classes $A = \{a_1, a_2, \ldots, a_n\}$, and W_s be a random variable which range over the set of video-words $W_s = \{w_1, w_2, \ldots, w_m\}$ at pyramid level s. Then the information about A captured

by W_s can be expressed by the Mutual Information (MI), $I(A, W_s)$. Now, let $\widehat{W_s} = \{\hat{w_1}, \hat{w_2}, \ldots, \hat{w_k}\}$ for $k < m$, be the compressed video-word cluster of W_s. We can measure the loss of quality of the resulting compressed vocabulary $\widehat{W_s}$, as the loss of MI:

$$Q(\widehat{W_s}) = I(A, W_s) - I(A, \widehat{W_s}) \tag{6.12}$$

To find the optimal compression $\widehat{W_s}$ we use Agglomerative Information Bottleneck (AIB) [54]. We use the described vocabulary compression at each level of the pyramid per class, and obtain a final class-specific compact pyramid representation of video-words.

6.4.3 Action Classification

After compression of the video-words at each pyramid level we compute a histograms of the video-words, using the extracted HOF3D descriptors, and concatenate them to a final feature set for SVM learning. We design a class specific χ-square kernel-based SVM, $\text{SVM}_{a_i}(k, h^{a_i}_{W_{a_i}})$ [6]. Where a_i is the i^{th} action class A, k is the SVM kernel and $h^{a_i}_{W_{a_i}}$ is the histogram of action class a_i, computed using the class-specific video-words W_{a_i}. For a test set a_{Test} we detect its action class:

$$i^*_{a_{Test}} = argmax_j \text{SVM}_{a_j}(k, h^{a_{Test}}_{W_{a_j}}), \forall a_j \in A \tag{6.13}$$

6.5 Experimental Results

To test our proposed approach we conduct a number of experiments: (1) action recognition using publicly available multi-view datasets and comparison with the state-of-the-art, (2) an comparison of the different variants of the HOF3D descriptor and 3D spatial pyramids, (3) an incremental analysis of the performance of the vocabulary building process, and (4) evaluation of view-invariance using different camera views for training and testing of the system.

6.5.1 Datasets

We evaluate our approach using the publicly available dataset: i3DPost Multi-View Human Action Dataset[1] [12]. and the INRIA Xmas Motion Acquisition Sequences (IXMAS) Multi-View Human Action Dataset[2] [62].

i3DPost

The i3DPost dataset consists of 8 actors performing 10 different actions, where 6 are single actions: *walk*, *run*, *jump*, *bend*, *hand-wave* and *jump-in-place*, and 4 are combined actions: *sit-stand-up*, *run-fall*, *walk-sit* and *run-jump-walk*. The subjects have different body sizes, clothing and are of different sex and nationalities. The multi-view videos

[1]The i3DPost dataset is available at http://kahlan.eps.surrey.ac.uk/i3dpost_ action/data

[2]The IXMAS dataset is available at http://4drepository.inrialpes.fr/public/ viewgroup/6

Figure 6.8: Image and 3D mesh model examples for the 10 actions from the i3DPost Multi-View Human Action Dataset. The figure is organized such that the columns correspond to the 10 different actions performed by the 8 actors, where the first 6 columns show the single actions and the last 4 columns show the combined actions. The first 8 rows depict images captured from the 8 camera views, while the 9^{th} row shows the corresponding 3D mesh models.

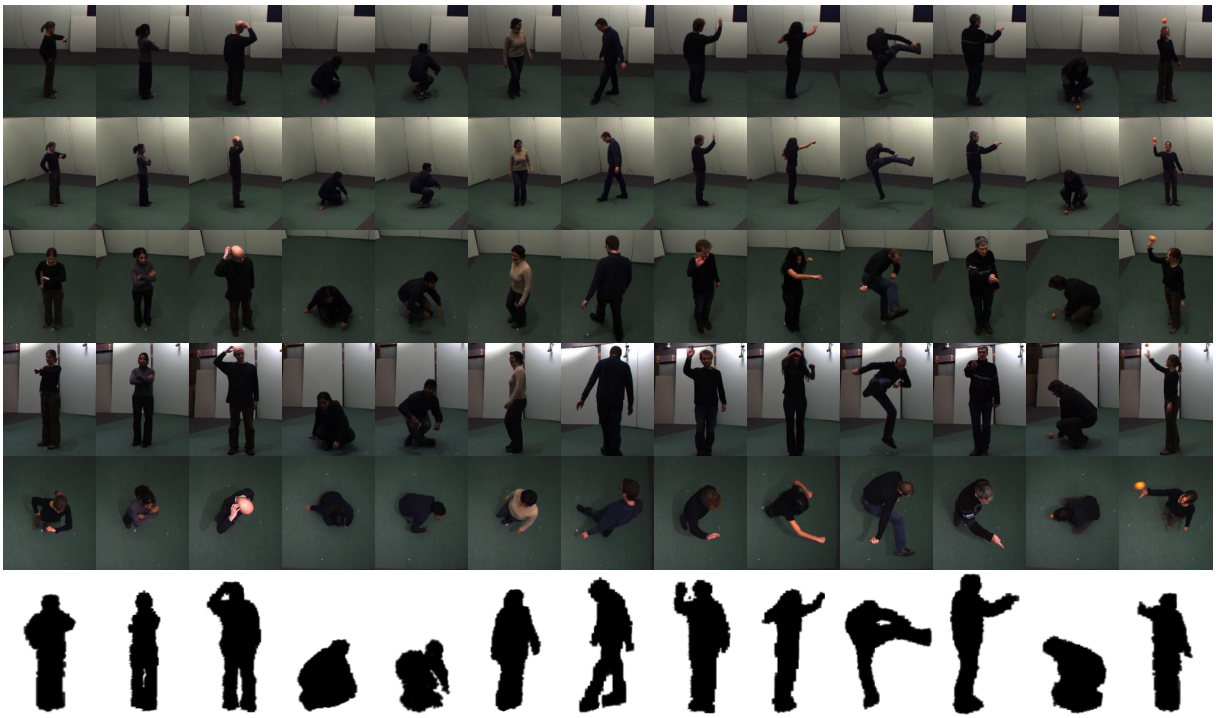


Figure 6.9: Image and 3D voxel-based volume examples for the 13 actions from the IXMAS Multi-View Human Action Dataset. The figure is organized such that the columns correspond to the 13 different actions performed by the 12 actors. The first 5 rows depict images captured from the 5 camera views, while the 6^{th} row shows the corresponding 3D volumes.

have been recorded by a 8 calibrated and synchronized camera setup in high definition resolution (1920×1080), resulting in a total of 640 videos. For each video frame a 3D mesh model of relatively high detail level ($20,000$-$40,000$ vertices and $40,000$-$80,000$ triangles) of the actor and the associated camera calibration parameters are available. The mesh models were reconstructed using a global optimization method proposed by Starck and Hilton [56]. Figure 6.8 shows multi-view actor/action images and 3D mesh model examples from the i3DPost dataset.

IXMAS

The IXMAS dataset consists of 12 non-professional actors performing 13 daily-life actions 3 times: *check watch*, *cross arms*, *scratch head*, *sit down*, *get up*, *turn around*, *walk*, *wave*, *punch*, *kick*, *point*, *pick up* and *throw*. The dataset has been recorded by 5 calibrated and synchronized cameras, where the actors chose freely position and orientation, and consists of image sequences (390×291) and reconstructed 3D volumes ($64 \times 64 \times 64$ voxels), resulting in a total of 2340 action instances for all 5 cameras. I.e, compared to i3Dpost the IXMAS dataset is of lower data quality and resolution. In the following we will show how our approach performs on both of these datasets. Figure 6.9 shows multi-view actor/action images and voxel-based volume examples from the IXMAS datasets.

Table 6.1: State-of-the-art recognition accuracies (%) for the i3DPost dataset. The column named "Dim" states if the methods apply 2D image data or 3D data. *Gkalelis et al. [13] test on 5 single actions.

Method	Dim	8 actions	10 actions
$HOF3D_{norm}$ + SP_{model}	3D	**98.44**	**97.50**
$HOF3D_{flow}$ + SP_{model}	3D	96.88	**97.50**
$HOF3D_{mode}$ + SP_{model}	3D	95.31	93.75
HHOF3D + SP_{model}	3D	93.75	95.00
$HOF3D_{norm}$ + SP_{STIPs}	3D	96.88	95.00
$HOF3D_{flow}$ + SP_{STIPs}	3D	**98.44**	96.25
$HOF3D_{mode}$ + SP_{STIPs}	3D	93.75	93.75
HHOF3D + SP_{STIPs}	3D	93.75	92.50
Holte et al. [16]	3D	92.19	78.75
Iosifidis et al. [20]	2D	90.88	-
Gkalelis et al. [13]	2D	90.00*	-

6.5.2 Evaluation on i3DPost

For the first test we use the data available for all 8 camera views and the full action set of 10 actions (single and combined). Additionally, we split the combined action up into two additional single actions [20], resulting in a total of 8 single actions. We perform leave-one-out cross validation, hence, we use one actor for testing, while the system is trained using the rest of the dataset. Table 6.1 presents the results of our approach using the described variants of the HOF3D descriptors and 3D spatial pyramids in comparison to Iosifidis et al. [20] and Gkalelis et al. [13]. The results show comparable performance for the descriptor and pyramid variants, but with a slightly better overall performance using $HOF3D_{norm}$ + SP_{model}, followed up by $HOF3D_{flow}$ + SP_{model} and $HOF3D_{flow}$ + SP_{STIPs}. For the 8 single actions, the accuracy of $HOF3D_{norm}$ + SP_{model} and $HOF3D_{flow}$ + SP_{STIPs} are **98.44**%, while for the full action set of 10 actions, the accuracy of $HOF3D_{norm}$ + SP_{model} and $HOF3D_{flow}$ + SP_{model} are **97.50**%. The other two descriptor variants, $HOF3D_{mode}$ and HHOF3D, have slightly lower but comparable performance. These results are consistent with our expectations, since HHOF3D is an approximation of HOF3D by decomposing the representation into spherical harmonic basis functions within a certain bandwidth, while the circular bin shifting variant $HOF3D_{mode}$ can be seen as a fast but more coarse vertical rotation. In general the 3D spatial pyramid divisions based on a horizontal plane estimated as the center of gravity of the 3D human model (SP_{model}) performs slightly better considering all descriptors variants. This might be due to better location and precision of the horizontal plane, compared to the one estimated as the center of gravity of the detected STIPs (SP_{STIPs}), which can variate due to the amount of detected STIPs.

Incremental Analysis

Next we conduct an incremental analysis to investigate the performance boost by applying the 3D spatial pyramids and vocabulary compression. Figure 6.10 shows the recognition accuracy for the four HOF3D variants with and without 3D spatial pyramids (SP_{model}

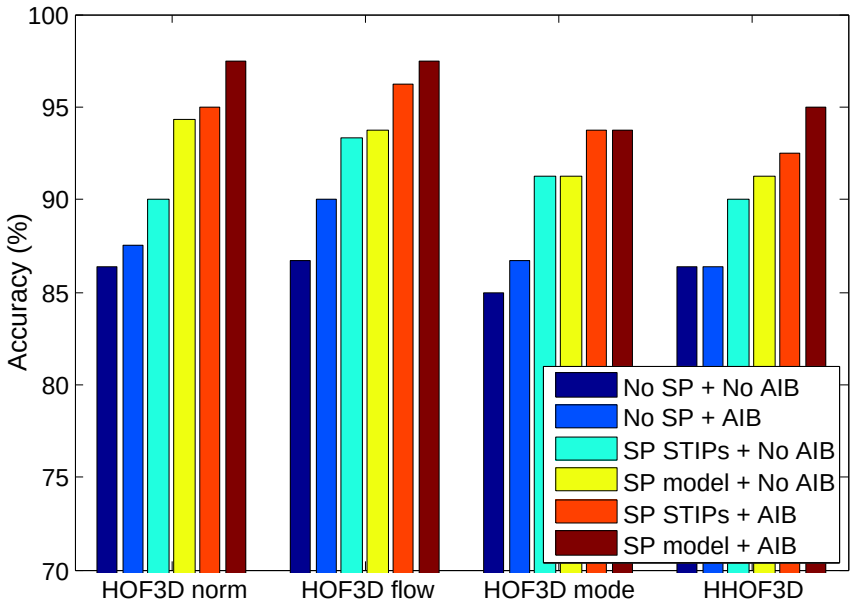


Figure 6.10: Plot of the recognition accuracy of the four HOF3D variants with and without spatial pyramids or AIB compression (i3DPost).

and SP_{STIPs}) or AIB vocabulary compression. The plot clearly indicates the performance boost by using spatial pyramids and compression for all descriptor variants. The largest performance increase occurs when applying spatial pyramids (∼5.5%). The vocabulary compression improves the average accuracy by ∼1.5%, however, when AIB is applied at pyramid level the performance boost is more significant (∼3%).

View-Invariance

To observe the view-invariance of our approach we evaluate its capability to recognize actions using different camera views for training and testing. We train and test the system by detecting STIPs, extracting $HOF3D_{norm}$ + SP_{model} descriptors and building vocabularies for classification for each of the 8 views, separately. Figure 6.11 shows a plot of the results, when recognizing all 10 actions using each combination of the 8 views for training and testing. As can be seen from the plot, the recognition accuracy is quite stable over all view combinations ($\sim 91\% \pm 6\%$). Note that only a small increase in the accuracy can be observed, when training and testing with the same view.

6.5.3 Evaluation on IXMAS

Table 6.2 presents the results of our approach using the HOF3D descriptors and 3D spatial pyramids (SP) in comparison to the state-of-the-art methods. Some authors only test on 11 actions performed by 10 actors (the test setup proposed by Weinland et al. [62]), while others evaluate their algorithms on the full dataset. Hence, to compare our approach to other works, we apply both test setups. As shown in the table our approach achieves a perfect recognition for both the 11 and 13 action setup, and thereby outperforms other proposed methods. The recognition accuracies are identical for all HOF3D descriptor and

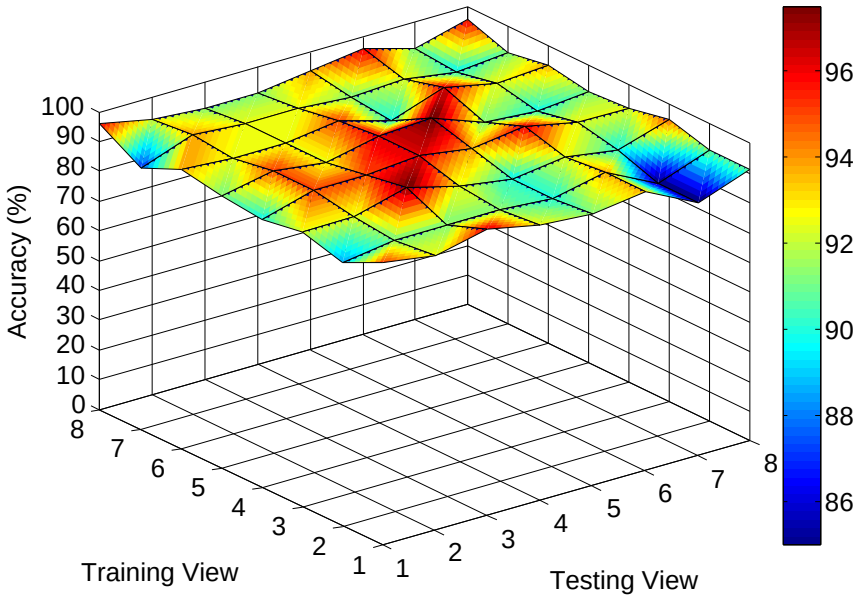


Figure 6.11: Plot of the recognition accuracy as a function of the applied camera views for training and testing (i3DPost).

pyramid variants. Futhermore, this validates that our approach can be used for multi-view data of lower data quality and resolution.

6.6 Conclusion

We have presented a 4D STIP and local 3D motion descriptor-based approach for human action recognition using 3D data acquired by multi-camera setups. We contribute to this field by: (1) the design of a 4D STIP detector, which operates in a selective manner by incorporating surround suppression and local spatio-temporal contraints. (2) Introducing a novel local 3D motion descriptor (HOF3D) for description of estimated 3D optical flow, and examine a number of solutions to make it view-invariant. (3) Based on 3D spatial pyramids of HOF3D descriptors we build a BoW vocabulary of human actions, which is compressed and classified using AIB and SVM, respectively. (4) We have reported superior performance on the publicly available i3DPost and IXMAS datasets, investigated the incremental performance boost of the proposed 3D spatial pyramids and vocabulary compression, and evaluated the view-invariance of the approach.

In future work it would be interesting to adapt the method to single view depth sensors (Time-of-Flight range cameras and the Kinect sensor [52]), which in general are are more flexible and applicable. Multi-camera systems are limited to a specific area of interest, due to its nature. However, it also helps to uncover occluded action regions from different views in the global 3D data, and allows for extraction of informative features in a more rich 3D space, than the one captured from a single view.

Table 6.2: State-of-the-art recognition accuracies (%) for the IXMAS dataset. The column named "Dim" states if the methods apply 2D image data or 3D data.

Method	Dim	11 actions	13 actions
HOF3D + SP	3D	**100.00**	**100.00**
Turaga et al. [59]	3D	98.78	-
Weinland et al. [62]	3D	93.33	-
Pehlivan et al. [47]	3D	90.91	88.63
Vitaladevuni et al. [61]	2D	87.00	-
Haq et al. [14]	2D	83.69	-
Weinland et al. [63]	2D	83.50	-
Liu et al. [39]	2D	-	82.80
Liu et al. [40]	2D	82.80	-
Weinland et al. [64]	2D	81.27	-
Lv et al. [43]	2D	-	80.60
Tran et al. [58]	2D	-	80.22
Cherla et al. [7]	2D	-	80.05
Liu et al. [37]	2D	-	78.50
Yan et al. [68]	3D	78.00	-
Junejo et al. [25]	2D	74.60	-
Junejo et al. [24]	2D	72.70	-
Reddy et al. [50]	2D	-	72.60
Farhadi et al. [10]	2D	58.10	-

Acknowledgements

This work has been supported by the Danish National Research Councils - FTP under the research project "Big Brother *is* watching you!"; the Spanish Research Programs Consolider-Ingenio 2010:MIPRCV (CSD200700018); Avanza I+D ViCoMo (TSI-020400-2009-133); and the Spanish project TIN2009-14501-C02-02.

References

[1] M. Ahmad and S.-W. Lee. Hmm-based human action recognition using multiview image sequences. In *ICPR*, 2006.

[2] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. 3d shape histograms for similarity search and classification in spatial databases. In *SSD*, 1999.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *PAMI*, 24(4):509–522, 2002.

[4] A.F. Bobick and J.W. Davis. The recognition of human movement using temporal templates. *PAMI*, 23:257–267, 2001.

[5] B. Chakraborty, M.B. Holte and. T.B. Moeslund, and J. Gonzáles. A selective spatio-temporal interest point detector for human action recognition in complex scenes. In *ICCV*, 2011.

[6] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. http://www.csie.ntu.edu.tw/∼cjlin/ libsvm.

[7] S. Cherla, K. Kulkarni, A. Kale, and V. Ramasubramanian. Towards fast, view-invariant human action recognition. In *CVPR Workshops*, 2008.

[8] I. Cohen and H. Li. Inference of human postures by classification of 3d human body shape. In *AMFG*, 2003.

[9] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.

[10] A. Farhadi and M.K. Tabrizi. Learning to recognize activities from the wrong view point. In *ECCV*, 2008.

[11] Preben Fihl and Thomas B. Moeslund. Invariant gait continuum based on the duty-factor. *SIViP*, 3(4):391–402, 2008.

[12] N. Gkalelis, H. Kim, A. Hilton, N. Nikolaidis, and I. Pitas. The i3dpost multi-view and 3d human action/interaction database. In *CVMP*, 2009.

[13] N. Gkalelis, N. Nikolaidis, and I. Pitas. View indepedent human movement recognition from multi-view video exploiting a circular invariant posture representation. In *ICME*, 2009.

[14] Anwaar Haq, Iqbal Gondal, and Manzur Murshed. On dynamic scene geometry for view-invariant action matching. In *CVPR*, 2011.

[15] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.

[16] M.B. Holte, T.B. Moeslund, N. Nikolaidis, and I. Pitas. 3d human action recognition for multi-view camera systems. In *3DIMPVT*, 2011.

[17] P. Huang and A. Hilton. Shape-colour histograms for matching 3d video sequences. In *3DIM*, 2009.

[18] P. Huang, A. Hilton, and J. Starck. Shape similarity for 3d video sequences of people. *IJCV*, 89:362–381, 2010.

[19] P. Huang, J. Starck, and A. Hilton. A study of shape similarity for temporal surface sequences of people. In *3DIM*, 2007.

[20] A. Iosifidis, N. Nikolaidis, and I. Pitas. Movement recognition exploiting multi-view information. In *MMSP*, 2010.

[21] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007.

[22] X. Ji and H. Liu. Advances in view-invariant human motion analysis: A review. *Trans. Sys. Man Cyber Part C*, 40(1):13–24, 2010.

[23] A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *PAMI*, 21(5):433–449, 1999.

[24] I.N. Junejo, E. Dexter, I. Laptev, and P. Pérez. Cross-view action recognition from temporal self-similarities. In *ECCV*, 2008.

[25] I.N. Junejo, E. Dexter, I. Laptev, and P. Pérez. View-independent action recognition from temporal self-similarities. *PAMI*, 33(1):172–185, 2011.

[26] I.-K. Jung and S. Lacroix. A robust interest points matching algorithm. In *ICCV*, 2001.

[27] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *SGP*, 2003.

[28] A. Kläser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008.

[29] J.J. Koenderink and A.J. Van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.

[30] M. Körtgen, M. Novotni, and R. Klein. 3d shape matching with 3d shape contexts. In *CESCG*, 2003.

[31] I. Laptev. On space-time interest points. *IJCV*, 64(2/3):107–123, 2005.

[32] I. Laptev, B. Caputo, C. Schüldt, and T. Lindeberg. Local velocity-adapted motion events for spatio-temporal recognition. *IJCV*, 108(3):207–229, 2007.

[33] I. Laptev and T. Lindeberg. Local descriptors for spatio-temporal recognition. In *First International Workshop on Spatial Coherence for Visual Motion Analysis*, 2004.

[34] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[35] Wanqing Li, Zhengyou Zhang, and Zicheng Liu. Action recognition based on a bag of 3d points. In *CVPR Workshops*, 2010.

[36] T. Lindeberg. Feature detection with automatic scale selection. *IJCV*, 30(2):79–116, 1998.

[37] J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. In *CVPR*, 2008.

[38] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *CVPR*, 2009.

[39] J. Liu and M. Shah. Learning human actions via information maximization. In *CVPR*, 2008.

[40] Jingen Liu, Mubarak Shah, Benjamin Kuipers, and Silvio Savarese. Cross-view action recognition via view knowledge transfer. In *CVPR*, 2011.

[41] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[42] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Imaging Understanding Workshop*, 1981.

[43] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *CVPR*, 2007.

[44] T.B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 104(2-3):90–126, 2006.

[45] A. Oikonomopoulos, I. Patras, and M. Pantic. Spatiotemporal salient points for visual recognition of human actions. *SMC-B*, 36(3):710–719, 2006.

[46] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Trans. Graph.*, 21:807–832, 2002.

[47] S. Pehlivan and P. Duygulu. A new pose-based representation for recognizing actions from multiple cameras. *CVIU*, 115:140–151, 2011.

[48] M. Pierobon, M. Marcon, A. Sarti, and S. Tubaro. 3-d body posture tracking for human action template matching. In *ICASSP*, 2006.

[49] Ronald Poppe. A survey on vision-based human action recognition. *IVC*, 28(6):976–990, 2010.

[50] K.K. Reddy, J. Liu, and M. Shah. Incremental action recognition using feature-tree. In *ICCV*, 2009.

[51] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ACM International Conference on Multimedia*, 2007.

[52] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011.

[53] L. Sigal and M.J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. In *Techniacl Report*, 2006.

[54] N. Slonim and N. Tishby. Agglomerative information bottleneck. In *NIPS*, 1999.

[55] R. Souvenir and J. Babbs. Learning the viewpoint manifold for action recognition. In *CVPR*, 2008.

[56] J. Starck and A. Hilton. Surface capture for performance based animation. *IEEE Computer Graphics and Applications*, 27(3):21–31, 2007.

[57] A. Swadzba, N. Beuter, J. Schmidt, and G. Sagerer. Tracking objects in 6d for reconstructing static scenes. In *CVPR Workshops*, 2008.

[58] D. Tran and A. Sorokin. Human activity recognition with metric learning. In *ECCV*, 2008.

[59] P. Turaga, A. Veeraraghavan, and R. Chellappa. Statistical analysis on stiefel and grassmann manifolds with applications in computer vision. In *CVPR*, 2008.

[60] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. *PAMI*, 27(3):475–480, 2005.

[61] S.N. Vitaladevuni, V. Kellokumpu, and L.S. Davis. Action recognition using ballistic dynamics. In *CVPR*, 2008.

[62] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *CVIU*, 104(2):249–257, 2006.

[63] Daniel Weinland, Mustafa Özuysal, and Pascal Fua. Making action recognition robust to occlusions and viewpoint changes. In *ECCV*, 2010.

[64] Daniel Weinland, Rémi Ronfard, and Edmond Boyer. Action recognition from arbitrary views using 3d exemplars. In *ICCV*, 2007.

[65] Daniel Weinland, Rémi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *INRIA Report*, RR-7212:54–111, 2010.

[66] G. Willems, T. Tuytelaars, and L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, 2008.

[67] S.F. Wong and R. Cipolla. Extracting spatiotemporal interest points using global information. In *ICCV*, 2007.

[68] P. Yan, S.M. Khan, and M. Shah. Learning 4d action feature models for arbitrary view action recognition. In *CVPR*, 2008.

[69] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. In *CVPR*, 2005.

Chapter 7

Foreground Segmentation and Shadow Detection

This chapter consists of the paper "Shadow Tracking for Improved Detection and Removal of Chromatic Moving Shadows" [A]. The paper presents work on foreground segmentation and shadow detection using a multi-stage approach. The shadow detection is enhanced by tracking both foreground objects and shadow using mutual information. Reference [B] describes intermediate work resulting in the final outcome in [A].

References

A. I. Huerta, M.B. Holte, T.B. Moeslund and J. Gonzàlez. Shadow Tracking for Improved Detection and Removal of Chromatic Moving Shadows. Submitted to *Transactions on Image Processing, IEEE Signal Processing Society*, 2011.

B. I. Huerta, M.B. Holte, T.B. Moeslund and J. Gonzàlez. Detection and Removal of Chromatic Moving Shadows in Surveillance Scenarios. In *IEEE International Conference on Computer Vision, Kyoto, Japan*, September 2009.

Shadow Tracking for Improved Detection and Removal of Chromatic Moving Shadows

I. Huerta, M.B. Holte, T.B. Moeslund and J. Gonzàlez

Abstract

Segmentation in the surveillance domain has to deal with shadows to avoid distortions when detecting moving objects. Most approaches for shadow detection are typically restricted to penumbra shadows, hence, such techniques cannot cope well with umbra shadows. Consequently, umbra is usually detected as part of moving objects. In this paper, firstly a bottom-up approach using a novel technique based on gradient and colour models, for separating chromatic moving cast shadows from detected moving objects, is presented. Secondly, a top-down approach based on a tracking system has been developed, in order to enhance the chromatic shadow detection. In the bottom-up step, both a chromatic invariant colour cone model and an invariant gradient model are built to perform automatic segmentation while detecting potential shadows. Regions corresponding to potential shadows are grouped by considering the "bluish effect" and an edge partitioning. Additionally, (i) temporal similarities between textures and (ii) spatial similarities between chrominance angle and brightness distortions are analysed for all potential shadow regions, in order to finally identify umbra shadows. In the top-down process, after detection of objects and shadows, both are tracked using Kalman filters. This implies data association between the blobs (foreground and shadow) and Kalman filters using Probabilistic Appearance Models. Based on an event analysis, we are testing for temporal consistency in the association between objects and shadows and their respective Kalman Filters. The results of tracking are used as feedback to recover miss-detected shadows. Unlike other methods, our approach does not make any a-priori assumptions about camera location, surface geometries, surface textures, shapes and types of shadows, objects, and background. Experimental results show state-of-the-art performance for different shadowed materials and illumination conditions.

7.1 Introduction

A fundamental problem for all automatic video surveillance systems is to detect objects of interest in a given scene. A commonly used technique for segmentation of moving objects is background subtraction [13]. This involves detection of moving regions (i.e., the foreground) by differencing the current image and a reference background image in a pixel-by-pixel manner. An important challenge for foreground segmentation is the impact of shadows. Shadows can be divided into two categories: *static shadows* and *dynamic (moving) shadows*. Static shadows occur due to static background objects (e.g., trees, buildings, parked cars, etc.) blocking the illumination from a light source. Static shadows can be incorporated into the background model, while dynamic shadows have shown to be more problematic. Dynamic shadows are due to moving objects (e.g., people, vehicles, etc.). The impact of dynamic shadows can be crucial for the foreground segmentation, and cause objects to merge, distort their size and shape, or occlude other objects. This results in a reduction of computer vision algorithms' applicability for, e.g, scene monitoring, object recognition, target tracking and counting.

Dynamic shadows can take any size and shape, and can be both *umbra* (dark shadow) and *penumbra* (soft shadow) shadows. Penumbra shadows exhibit low values of intensity but similar chromaticity values w.r.t. the background, while umbra shadows can exhibit different chromaticity than the background, and their intensity values can be similar to those of any new object appearing in a scene. When the chromaticity of umbra shadows differs from the chromaticity of the global background illumination, we define this as *chromatic shadow*. Consequently, umbra shadows are significantly more difficult to detect, and therefore usually detected as a part of moving objects. When a shadow has successfully been detected it is usually removed instantly, since only the object is of interest for further processing and not the shadow. As a result, the shadow information is lost. Our idea is to use this information to improve other aspects of object and shadow detection and tracking. Concretely, if a detected shadow is tracked over time instead of being discarded, it could be used to improve the shadow detection and possible the object detection and tracking as well.

Shadow detection is an important field of research within computer vision. Even though many algorithms have been proposed, the problem of detection and removal of shadows in complex environment is still far from being completely solved. A common direction is to assume that shadows decrease the luminance of an image, while the chrominance stays relatively unchanged [1, 10]. However, this is not the case in many scenarios, e.g., in outdoor scenes. Other approaches apply geometrical information. Onoguchi [15] uses two cameras to eliminate the shadows of pedestrians based on object height, where objects and shadows must be visible to both cameras. Ivanov et al. [9] apply a disparity model, which is invariant to arbitrarily rapid changes in illumination, for modelling background. However, to overcome rapid changes in illumination at least three cameras are required. In [19], Salvador et al. exploit the fact that a shadow darkens the surfaces, on which it is cast, to identify an initial set of shadowed pixels. This set is then pruned by using colour invariance and geometric properties of shadows. It should be noted that most of the approaches which applies geometrical information normally requires shadows to be cast on a flat plane.

Another popular approach is to exploit colour differences between shadow and background in different colour spaces. In [2], Cucchiara et al. consider the hypothesis that shadows reduce surface brightness and saturation while maintaining the hue properties in the HSV colour space. Schreer et al. [20] adopt the YUV colour space, while Horprasert et al. [6], Kim et al. [10] and [16] build a model in the RGB colour space to express normalised luminance variation and chromaticity distortions. However, these methods require illumination sources to be white, and assume shadow and non-shadow have similar chrominance. Some authors use textures to obtain a segmentation without shadows, e.g, Heikkila et al. [5] apply Local Binary Patterns, but it also fails to detect umbra shadows.

To overcome these shortcomings, a number of approaches apply colour constancy methods, combine different techniques or use multi-stage approaches. A comparative study of shadow detection techniques can be found in [17]. In addition to scene brightness properties, Stauder et al. [22] extract edge width information to differentiate penumbra regions from the background. In [3], Finlayson et al. use shadow edges along with illuminant invariant images to recover full colour shadow-free images. Nonetheless, a part of the colour information is lost in removing the effect of the scene illumination at each pixel in the image. Weiss [23] computes the reflectance edges of the scene to obtain an intrinsic image without shadows. However, this approach requires significant changes in the scene, and the reflectance image also contains the scene illumination. Martel et al. introduce a parametric approach based on Gaussian mixtures GMSM [17]. Additionally, they propose a nonparametric framework based on the physical properties of light sources and surfaces, and apply spatial gradient information to reinforce the learning of model parameters [18]. Finally, [14] proposes a multi-stage approach for outdoor scenes, which is based on a spatio-temporal albedo test and dichromatic reflection model.

In this paper, firstly a bottom-up approach for detection and removal of chromatic moving shadows in surveillance scenarios is presented [8]. We apply a multi-stage approach inspired by [14] but we use multiple cues: colour and gradient information, together with known shadow properties. Secondly, a top-down architecture based on a tracking system is proposed in order to enhance the chromatic shadow detection, where Kalman filters are used for tracking. Shadows can be lost for a number of frames of a video sequence, and in these cases the use of Kalman filters to track the shadows can improve the shadow detection. Fig. 7.1 illustrates a high level scheme for our shadow detection and tracking approach.

In contrast to the aforementioned approaches, this paper contains the following contributions: (i) we combine an invariant colour cone model and an invariant gradient model to improve foreground segmentation and detection of potential shadows. (ii) We extend the shadow detection to cope with chromatic moving cast shadows, by grouping potential shadow regions and considering the " bluish effect", edge partitioning, temporal similarities between local gradient, and spatial similarities between chrominance angle and brightness distortions. (iii) We track both objects and shadows, and thereby establishing data association between them. Hereby, an enhancement of the chromatic shadow detection is achieved by recovering miss-detected shadows. To the best of our knowledge, we are the first to apply shadow tracking for improving object and shadow detection in surveillance. As a result we obtain: (iv) a more robust tracking by using mutual information and association of object and shadow, and (v) improvement of the segmentation for high

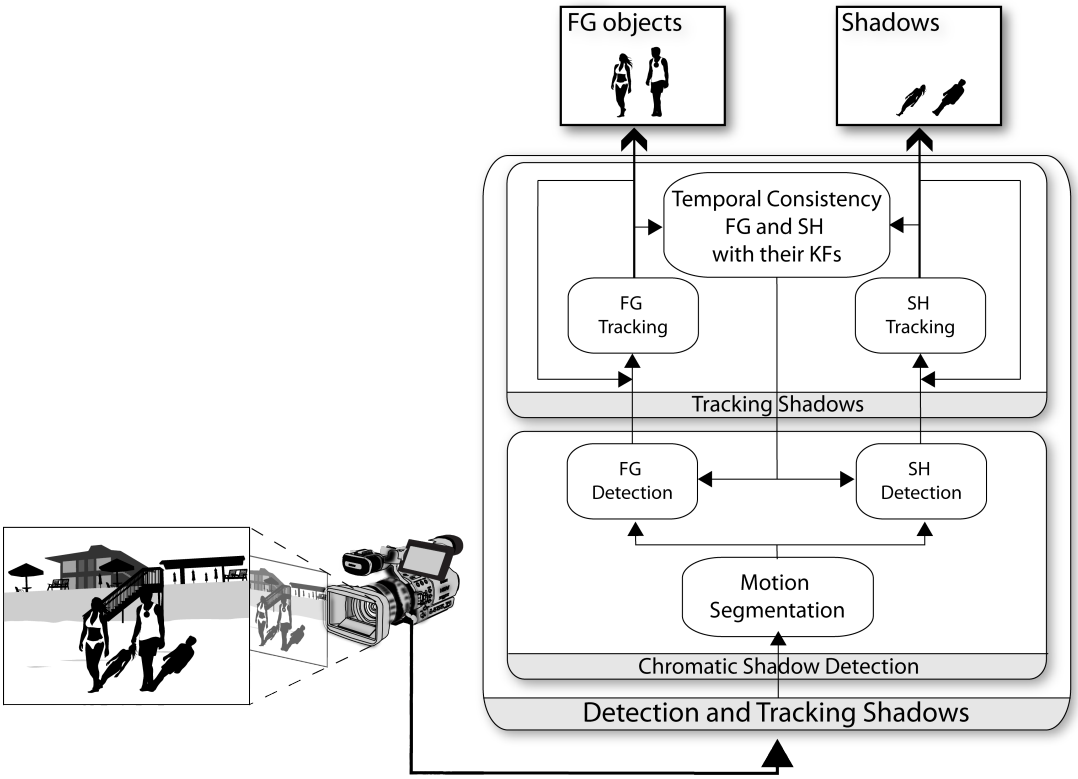


Figure 7.1: Flowchart for the shadow detection and tracking system.

level processes, such as detection and tracking, by avoiding shadows. (vi) Unlike other methods, our approach does not make any assumptions about camera location, surface geometries, surface textures, shapes and types of shadows, objects and background.

The remainder of the paper is organised as follows. In section 7.2, the theoretical concept of our approach is outlined. The algorithm for foreground segmentation, along with the detection and removal of chromatic moving shadows are described in section 7.3. The top-down process used to enhance the shadow detection is described in section 7.4. Finally, we present experimental results in section 7.5 and concluding remarks in section 7.6.

7.2 Analysis of Shadow Properties

The colour information ρ at a given pixel a obtained from a recording camera supposing Lambertian surfaces depends on four components: the Spectral Power Distribution (SPD) of the illuminant denoted $E(\lambda)$, the surface reflectance $R(\lambda)$, the sensor spectral sensitivity $Q(\lambda)$ evaluated at each pixel a and a shading factor σ.

$$\rho_a = \sigma \int E(\lambda)R(\lambda)Q_a(\lambda)d\lambda \tag{7.1}$$

The surface reflectance $R(\lambda)$ depends on the material, i.e., materials have different response to the same illumination.

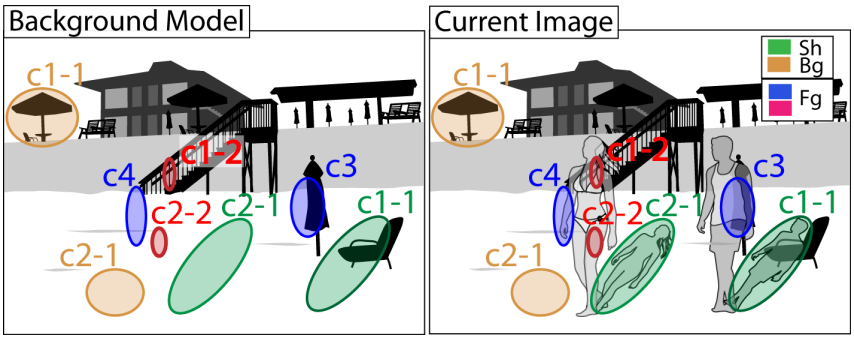


Figure 7.2: A sketch of the four main cases (c1 to c4: blue ellipses) and two anomalies (c1-2 and c2-2: red ellipses) that can occur, when performing foreground segmentation with the influence of shadows, using the temporal local gradients. The ellipses represent detection of potential chromatic shadows. They are grouped by considering an intensity reduction, "the bluish effect" and an edge partition.

7.2.1 The bluish effect

In outdoor scenes, the environment is illuminated by two light sources: a point light source (the sun) and a diffuse source (the sky) with different SPD $E(\lambda)$. Besides a reduction in the intensity, an outdoor cast shadow result in a change of the chrominance. The illumination of the sky has higher power components in the lower wavelengths λ (450 - 495 nm) of the visible spectrum, and it is therefore assumed bluish as argued in [14]. When the direct illumination of the sun is blocked and an region is only illuminated by the diffuse ambient light of the sky, materials appears to be more bluish.

7.2.2 Temporal local gradient information

By applying gradient information we can obtain knowledge about object boundaries, and thereby improve the foreground segmentation. Additionally, the gradient provides textural information about both the background and foreground image. Although shadows result in a intensity reduction of the illumination, and the texture of a given object or the background has lower gradient magnitude, the structure remains the same, i.e., the gradient orientation is unchanged.

7.2.3 Shadow scenaria and solutions

When performing foreground segmentation with the influence of shadows, and taking the temporal local gradients into account, four main cases can occur as illustrated in Fig. 7.2. The ellipses represent detection of potential chromatic shadows. They are grouped by considering an intensity reduction, "the bluish effect" and an edge partition.

***Case1*:** Similar local gradient structures are present in the background model and in the current image. By examining similarities between the local gradients, and the fact that there is no foreground object in the current image, potential shadows can be detected and identified as shadow regions (*case1-1*). However, if a foreground object

is present, it can be miss-classified as shadow if the gradients of the background and the foreground object are similar (*case1-2*).

***Case2*:** There is no available background model nor local gradients in the current image. Since, the change in illumination of all the potential shadow regions has to be similar, temporal and spatial similarities between chrominance angle and brightness distortions within the potential regions are analysed to detect chromatic shadows (*case2-1*). However, a foreground object can be miss-classified as shadow if the foreground object has no gradients. Furthermore, the chrominance angle distortion can also be similar among the pixels in the region of the object (*case2-2*).

***Case3*:** Local gradient structure is present in the background model but not in the current image. By examining similarities between temporal gradients, a potential shadow can be detected as a foreground object, if there are background gradients and a new foreground object in the current image.

***Case4*:** Local gradient structure is present in the current image but not in the background model. Then there must be a new foreground object in the current image. In this case, the gradients in the current image are employed for object detection. Hence, there is no need to analyse the potential region further.

The described characteristics are not sufficient to address the anomalies in *case1-2* and *case2-2*. Therefore, we take further assumptions and apply some additional steps, which are explained next.

7.3 Bottom-Up Chromatic Shadow Detection

Our approach, depicted in Fig. 7.3 is a multi-stage approach. The first three stages remove the pixels which cannot be shadow. The fourth step divide the regions of potential shadows. Chromatic shadow detection is performed in stage 5 and 6 based on gradients and chrominance angles, respectively. The last step avoids foreground regions to be erroneously detected as chromatic shadows. An example is given in Fig. 7.4.

7.3.1 Moving foreground segmentation

In this stage foreground objects, shadows and some erroneous pixels are segmented. In order to achieve moving foreground segmentation an improved hybrid approach based on [7], which fuses colour and gradient information, is used. Note that this approach can cope with several motion segmentation challenges, e.g., penumbra shadows, since it is based on a chromatic colour model [6]. We use similar architecture and automatic threshold selection as the hybrid approach in [7]. This architecture provides the highest detection rate in comparison to other motion segmentation approaches. However, the colour and gradient models are modified, in order to achieve a more accurate segmentation and to become applicable for the next stages.

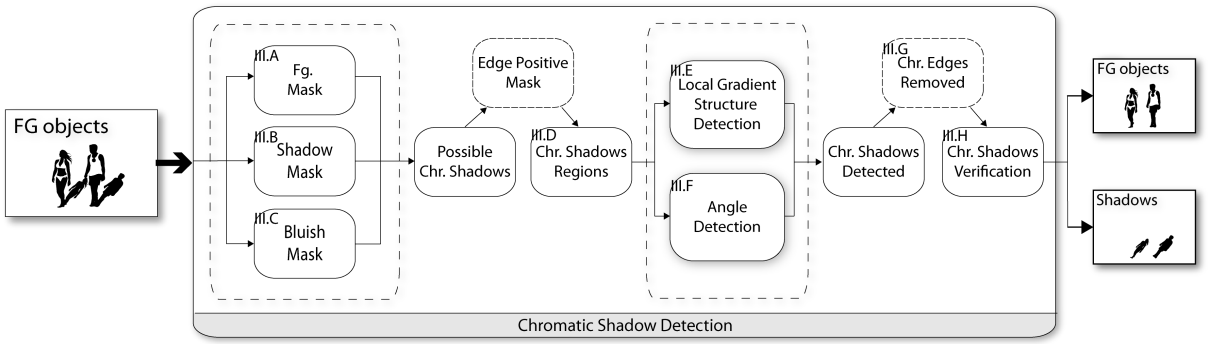


Figure 7.3: A schematic overview of the chromatic shadow detection approach.

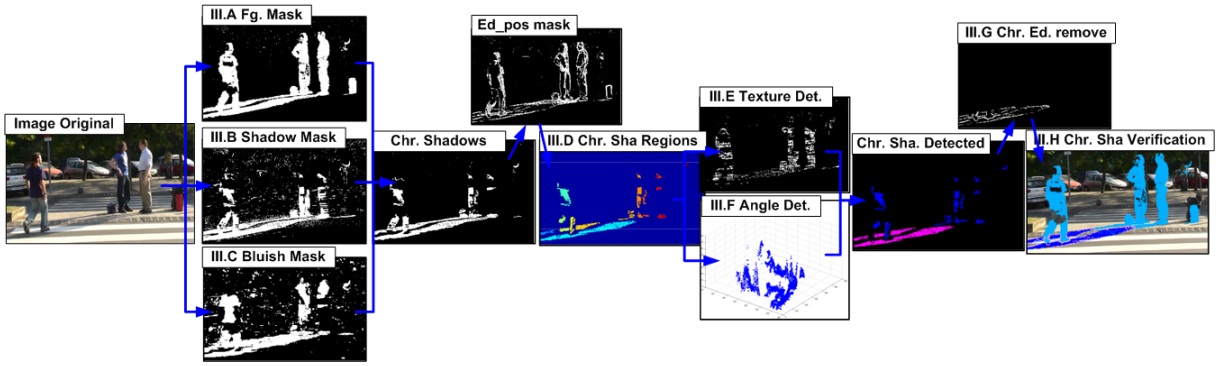


Figure 7.4: An example of chromatic shadow detection. The numbers added to the image captions correspond to the respective sub-sections in section 7.3.

The chromatic cylinder model employed in several motion segmentation approaches [10, 6, 7] is changed into a new chromatic cone model. It is based on chrominance angle distortion instead of chromatic distortion. For a chromaticity line, the chromatic distortion applied in the chromatic cylinder model depends on the brightness distortion, while the chrominance angle distortion is invariant to the brightness, as it can be seen in Fig. 7.5 (the chromatic distortion δ increases proportional to the brightness distortion α, while the chrominance angle distortion β is unaffected). The invariant chromatic cone model is more robust to chromatic shadows, since these (umbra) shadows modify both the brightness and the chromaticity. As argued in [12, 18], the gradient model has to be invariant to global and local illuminations changes, i.e., shadows. The new invariant gradient model presented in this section uses a combination of gradient magnitudes and gradient directions, which is invariant to illumination changes, and can be applied to identify the local gradient structures of an image.

Invariant colour cone model

The Background Colour Model (BCM) is computed according to the chromatic invariant cone representation shown in Fig. 7.5. First, the RGB mean $\boldsymbol{\mu}_a = \left(\mu_a^R, \mu_a^G, \mu_a^B\right)$ and standard deviation $\boldsymbol{\sigma}_a = \left(\sigma_a^R, \sigma_a^G, \sigma_a^B\right)$ of every image pixel a during the time period $t = [1 : T_1]$ are computed. Once each RGB component is normalised by their respective standard deviation σ_a^c (where $c \in \{R, G, B\}$ denotes the colour channel), two distortion

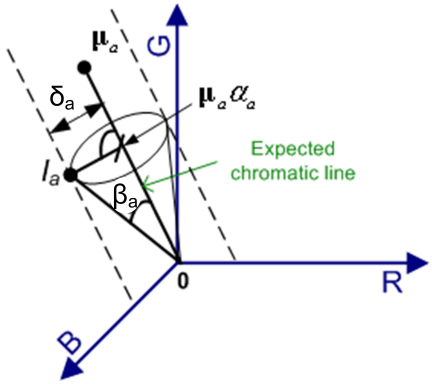


Figure 7.5: A colour cone model, where $\boldsymbol{\mu}_a$ represents the expected RGB colour value for a pixel a, and $\mathbf{I}_a$ is the current pixel value. The line $\overline{\mathbf{0}\boldsymbol{\mu}_a}$ shows the expected chromatic line, and all colours along this line have the same chrominance but different brightness. α_a and β_a give the current brightness and chrominance angle distortion, respectively.

measures are established during the training period: the brightness distortion, $\alpha_{a,t}$, and the chrominance angle distortion, $\beta_{a,t}$. The brightness distortion can be computed by minimising the distance between the current pixel value $\mathbf{I}_{a,t}$ and the chromatic line $\overline{\mathbf{0}\boldsymbol{\mu}_a}$. The angle between $\overline{\mathbf{0}\boldsymbol{\mu}_a}$ and $\overline{\mathbf{0}\boldsymbol{I}_a}$ is the chrominance angle distortion. Hence, the brightness and the chrominance angle distortions are given by:

$$\alpha_{a,t} = \frac{\frac{I^R_{a,t}\mu^R_a}{(\sigma^R_a)^2} + \frac{I^G_{a,t}\mu^G_a}{(\sigma^G_a)^2} + \frac{I^B_{a,t}\mu^B_a}{(\sigma^B_a)^2}}{\left(\frac{\mu^R_a}{\sigma^R_a}\right)^2 + \left(\frac{\mu^G_a}{\sigma^G_a}\right)^2 + \left(\frac{\mu^B_a}{\sigma^B_a}\right)^2} \tag{7.2}$$

$$\beta_{a,t} = \arcsin \frac{\sqrt{\sum_{c=R,G,B} \left(\frac{I^c_{a,t} - \alpha_{a,t}\mu^c_a}{\sigma^c_a}\right)^2}}{\sqrt{\sum_{c=R,G,B} \left(\frac{I^c_{a,t}}{\sigma^c_a}\right)^2}} \tag{7.3}$$

Next, the Root Mean Square over time of both distortions $\bar{\alpha}_a$ and $\bar{\beta}_a$ are computed for each pixel:

$$\bar{\alpha}_a = RMS\left(\alpha_{a,t} - 1\right) = \sqrt{\frac{1}{T_1}\sum_{t=0}^{T_1}\left(\alpha_{a,t} - 1\right)^2} \tag{7.4}$$

$$\bar{\beta}_a = RMS\left(\beta_{a,t}\right) = \sqrt{\frac{1}{T_1}\sum_{t=0}^{T_1}\left(\beta_{a,t}\right)^2} \tag{7.5}$$

where 1 is subtracted from $\alpha_{a,t}$, so that the brightness distortion is distributed around zero: positive values represent brighter pixels, whereas negative values represent darker pixels,

w.r.t the learnt values. These values are used as normalising factors so that a single global threshold can be set for the entire image. This 4-tuple BCM = $(\boldsymbol{\mu}_a, \boldsymbol{\sigma}_a, \bar{\alpha}_a, \bar{\beta}_a)$ constitutes the pixel-wise colour background model.

Invariant gradient model

The Background Edge Model (BEM) is built as follows: first the Sobel edge operator is applied to each colour channel in the horizontal and vertical directions. This yields a horizontal $G^c_{x,a,t} = S_x * I^c_{a,t}$ and a vertical $G^c_{y,a,t} = S_y * I^c_{a,t}$ gradient image for each frame during the training period $t = [1:T]$. Next, the gradient of each background pixel is modelled using the gradient mean $\mu_{Gx,a} = (\mu^R_{Gx,a}, \mu^G_{Gx,a}, \mu^B_{Gx,a})$ and $\mu_{Gy,a} = (\mu^R_{Gy,a}, \mu^G_{Gy,a}, \mu^B_{Gy,a})$, and the gradient standard deviation $\sigma_{Gx,a} = (\sigma^R_{Gx,a}, \sigma^G_{Gx,a}, \sigma^B_{Gx,a})$ and $\sigma_{Gy,a} = (\sigma^R_{Gy,a}, \sigma^G_{Gy,a}, \sigma^B_{Gy,a})$ computed for all the training frames. Then, the magnitude and orientation of the gradient mean (μ_G and μ_θ) and the standard deviation (σ_G and σ_θ) are computed in order to build the background edge model:

$$\mu^c_{G,a} = \sqrt{(\mu^c_{Gx,a})^2 + (\mu^c_{Gy,a})^2}; \quad \mu^c_{\theta,a} = \arctan\left(\frac{\mu^c_{Gy,a}}{\mu^c_{Gx,a}}\right) \tag{7.6}$$

$$\sigma^c_{G,a} = \sqrt{(\sigma^c_{Gx,a})^2 + (\sigma^c_{Gy,a})^2}; \quad \sigma^c_{\theta,a} = \arctan\left(\frac{\sigma^c_{Gy,a}}{\sigma^c_{Gx,a}}\right) \tag{7.7}$$

resulting in the 4-tuple BEM = $(\mu^c_{G,a}, \sigma^c_{G,a}, \mu^c_{\theta,a}, \sigma^c_{\theta,a})$. The thresholds employed for the segmentation task are automatically computed for each model, as described in [7].

Image segmentation

The colour segmentation is achieved by following the rules in [7]. However, the edge segmentation is achieved based on the following premises:

i Illumination changes modify the gradient magnitude but not the gradient orientation.

ii The gradient orientation is not feasible where there are no edges.

iii An edge can appear in a place where there were no edges before.

Assuming the first two premises, the gradient orientations will be compared instead of the gradient magnitudes for pixels which have a minimum magnitude, in order to avoid false edges due to illumination changes:

$$F_\theta = \left((\tau^c_{e,a} < V^c_{G,a,t}) \wedge (\tau^c_{e,a} < \mu^c_{G,a})\right) \wedge (\tau^c_{\theta,a} < |V^c_{\theta,a,t} - \mu^c_{\theta,a}|) \tag{7.8}$$

For pixels satisfying the third premise, the gradient magnitudes are compared instead of the orientations:

$$F_G = \left(\neg\left((\tau^c_{e,a} < V^c_{G,a,t}) \wedge (\tau^c_{e,a} < \mu^c_{G,a})\right)\right) \wedge (\tau^c_{G,a} < |V^c_{G,a,t} - \mu^c_{G,a}|) \tag{7.9}$$

where $V^c_{\theta,a,t}$ and $V^c_{G,a,t}$ are the gradient orientation and magnitude for each pixel in the current image, respectively.

The invariant models provide a high detection rate in comparison to other motion segmentation approaches. After the initial detection, moving foreground objects, chromatic shadows and some isolated pixels are represented by a binary mask named *M1*. Similarly, a mask is created using the gradient model and divided into two masks (*Edneg* and *Edpos*), which are used for the next steps. The *Edneg* mask corresponds to the foreground pixels belonging to the background model, while the *Edpos* mask corresponds to the foreground pixels belonging to the current image. Furthermore, a third mask is created called *Edcom*, which contains the common edges detected in the background model and in the current image.

7.3.2 Shadow intensity reduction

In this step the *M1* mask is reduced to avoid pixels which cannot be shadows. A foreground pixel cannot be a shadowed pixel if it has a higher intensity than the background model. Hence, a new mask *M2* is created according to equation 7.10.

$$M2_{a,t} = (I^R_{a,t} < \mu^R) \wedge (I^G_{a,t} < \mu^G) \wedge (I^B_{a,t} < \mu^B) \tag{7.10}$$

where a corresponds to the pixel location in *M1*.

7.3.3 The bluish effect

The effect of illuminants, which are different than white lights, provokes chromaticity changes, since the intensity variates differently for each color channel. In outdoor sequences the main illuminants are the sky and the sun (any of them white illuminant). The sky is the only source of illumination on shadowed regions, and it is assumed to be bluish, as argued in [14]. Therefore, the intensity changes in the red and green channels are larger than in the blue channel. This knowledge is used to reduce the potential shadow region detected in the previous step:

$$M3_{a,t} = (I^R_{a,t} - \mu^R) > (I^B_{a,t} - \mu^B) \wedge (I^G_{a,t} - \mu^G) > (I^B_{a,t} - \mu^B) \tag{7.11}$$

where a corresponds to the pixel location in *M2*. Obviously, the bluish effect cannot be applied for indoor sequences.

7.3.4 Potential chromatic shadow regions

It is supposed that shadow regions have similar intensity change for each channel, since the illuminant is the same. However, different surfaces have different reflectance characteristics, hence, the intensity change depends on the surface material. Therefore, we

apply edges to describe region borders. Concretely, we build a new mask *M4* using the foreground edges detected in the current image (*Edpos*) to separate the potential shadow regions from the moving foreground objects:

$$M4_{a,t} = M3_{a,t} \wedge \neg Edpos_{a,t} \tag{7.12}$$

A minimum area morphology is applied in order to avoid smaller regions, which do not contain enough information for the subsequent steps of the shadow analysis.

7.3.5 Chromatic shadow gradient detection

Next, the temporal gradients of the regions in *M4* are analysed to identify, which case of the theoretical shadow analysis (see section 7.2) each of the regions complies with. A region will be considered a shadow if it complies with case 1. The negative foreground edges (*Edneg*) of the region are compared to the common foreground edges (*Edcom*), in order to test if the region is a shadow and avoid the anomaly case 1-2.

$$Tx_b = \left(\frac{\sum\limits_{a \in R_b} (R_b \wedge Edneg)}{|R_b \wedge Edtot|} \cdot k_n < \frac{\sum\limits_{a \in R_b} (R_b \wedge Edcom)}{|R_b \wedge Edtot|} \right) \tag{7.13}$$

where a is the pixel position; R_b is the evaluated region and b is the number of the region; $|R_b|$ denotes the number of pixels of region b; $|R_b \wedge Edtot|$ denotes the number of pixels representing the edges detected in the background model and the current image; k_n corresponds to a confidence region, which is equal to the probability of the region belongs to a shadow or a foreground object.

7.3.6 Chromatic shadow angle and brightness detection

In this step temporal and spatial similarities of the chrominance angle and brightness distortion for all pixels belonging to regions, which have so far not been classified as shadow, are analysed. A region will be considered a shadow if it complies with case 2. The only regions analysed in this stage are those without gradients, neither in the background model nor in the current image. If the pixels do not have a significant gradient, but have similar chrominance angle distortion and similar brightness distortion, the region is classified as shadow.

$$\begin{aligned} ABd_b = & \left(\left(\left(\frac{\sum\limits_{a \in R_b} (R_b \wedge Edneg)}{|R_b \wedge Edtot|} \wedge \frac{\sum\limits_{a \in R_b} (R_b \wedge Edcom)}{|R_b \wedge Edtot|} \right) = 0 \right) \vee \left(\frac{\sum\limits_{a \in R_b} (R_b \wedge Edtot)}{|R_b|} < k_t \right) \right) \\ \wedge & \quad (\sigma(R_b \wedge \breve{\alpha}) < k_a) \wedge \Big(\sigma(R_b \wedge \breve{\beta}) < k_b \Big) \end{aligned} \tag{7.14}$$

where σ is the standard deviation of $\breve{\alpha}$ and $\breve{\beta}$ which are the chrominance angle and brightness normalised distortions calculated for each pixel in the region R_b, respectively; k_t [1] is a confidence region to avoid noise gradients; k_a and k_b are minimum thresholds used to determine if the angle and brightness distortion are similar among the pixels of the evaluated region.

7.3.7 Chromatic shadow edge removal

Pixels of the potential shadow regions, which were neglected in section 7.3.4, since they were part of the *Edpos* mask, are included again in the new set of shadow regions.

7.3.8 Shadow position verification

A moving cast shadow is always caused by a moving foreground object. Therefore, in this section it is tested if a detected shadow has an associated foreground object, in order to avoid the anomaly in case 2-2. Only shadows detected in the chrominance angle and brightness distortion analysis (section 7.3.6) will be tested. During a training period T_2, the angles between the detected shadows and the foreground objects are calculated. Hereafter, the most probable angle obtained in the training period is used to discard detected shadows, which do not have any foreground object in its direction.

7.4 Top-Down Shadow Tracking

When a shadow has successfully been detected it is usually removed, since it is the object which is of interest for further processing. As a result, the shadow information is lost. Our idea is to use this information a posteriori, in order to improve the shadow detection when it fails (e.g., due to camouflage problems). Concretely, if a detected shadow is tracked over time instead of being discarded, it can be used to recover miss-detected shadows, and hereby improve the shadow detection.

In this section a top-down approach is applied to enhance the chromatic shadow detection using a Kalman Filter (KF) based tracking. Fig. 7.6 shows an overview of the top-down tracking process, and the complete algorithm is listed in Algorithm 6. Firstly, the tracking module tracks objects and shadows through the scene. As input, the tracking module receives a binary mask from the object and shadow detection described in the previous section, as illustrated in Fig. 7.7. In the following subsections the tracker is explained with special attention on data association, an event analysis and Probabilistic Appearance Models (PAMs) (sections 7.4.1, 7.4.2 and 7.4.3, respectively). The output of the tracker is a list of tracks for each object and shadow, and their mutual association, which is used as feedback to improve the object and shadow detection. Secondly, the association between objects and shadows is described and updated for the KFs (sec. 7.4.4). Thirdly, temporal consistency is investigated in the association between FG and SH blobs, and their assigned

[1] Empirically found constant. If more than 10% of the pixels are considered edges then it cannot be noise.

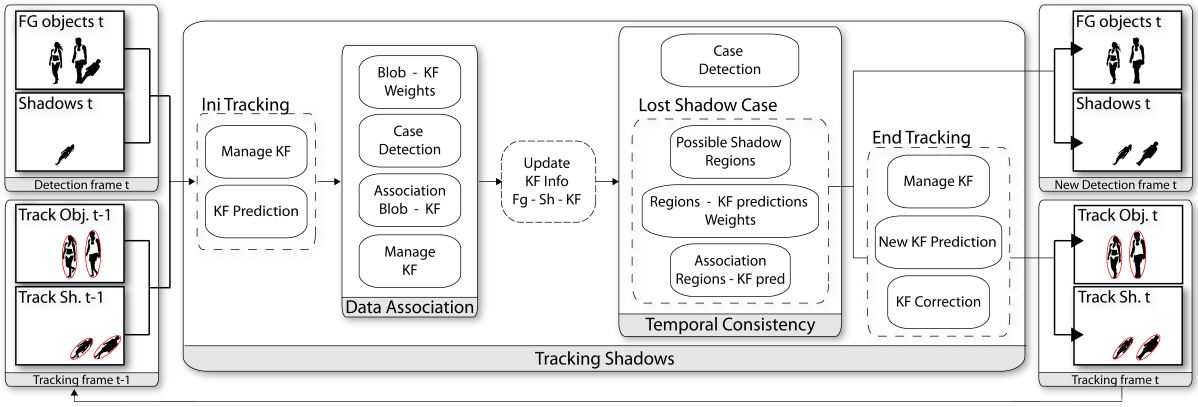


Figure 7.6: A schematic overview of the top-down shadow tracking process to enhance the chromatic shadow detection.

KFs, in order to identify possible lost shadows (sec. 7.4.5). Once the shadows are detected and tracked, the information is used as feedback to the chromatic shadow detection to recover miss-detected shadows in the original image (sec. 7.4.6). Finally, the KF and the PAM are updated, by taking the information from the new data association into account, and used for tracking in the next frames (sec. 7.4.7). An example of the entire process can be seen in the Fig. 7.7.

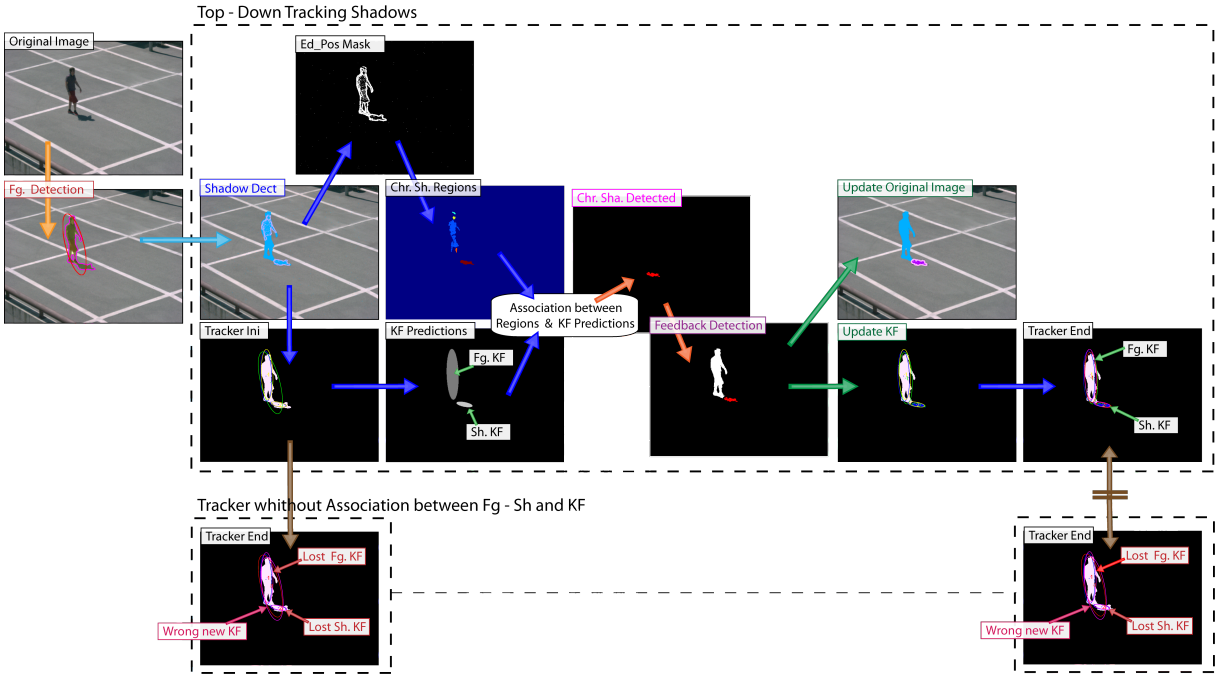


Figure 7.7: An example of the top-down tracking process. The figure illustrates the steps, within the tracking and the motion segmentation, applied to enhance the shadow detection, when a shadow is lost (lost shadow case). See the main body text for further details.

Algorithm 6 Top-down shadow tracking approach

For each blob from the chromatic shadow detection:

- Create a new Kalman Filter (KF) for each new blob and delete KFs, which have not been use in a period of time (Tdead).
- KF Prediction: Time update KF.
- Data Association between blobs (FG and SH) and KFs.
  - Build a Probabilistic Appearance Model (PAM) for each KF.
  - Compute weights for the association: two correspondence matrices using:
    - Euclidean distance based on the position (x,y) and the size (major and minor axes of an ellipse).
    - Matching of PAM and blob.
  - Case detection, see Algorithm 7:
    - Five possible cases: object match, new object, lost object, object splitting and object merging.
  - Establish association between the blobs (FG and SH) and the KFs.
  - Manage the KFs: updating, creating and deleting the KFs.
- Update KF: the association of the blobs and the KFs.
- Test for temporal consistency in the data association between FG, SH and their assigned KFs.
  - Case detection (see Fig. 7.10):
    - Three possible cases: FG and SH match, new shadow (FG-SH splitting) and lost shadow (FG-SH merging).
  - Lost Shadow case:
    - Detect possible shadow regions from the org. FG blob.
    - Compute weights for the association: two correspondence matrix using:
      - Euclidean distance based on the position (x,y) and the size (major and minor axes of an ellipse).
      - Matching of blobs.
    - Establish association between KF predictions of FG and SH, and the regions extracted from the org. FG blob.
- Feedback (top-down) from the tracking to the shadow detection:
  - Classify the original image using the data association and the new FG and SH blob information.
  - Update blob information for the original image.
- Manage the KFs: updating and deleting the KF:
  - Update the KF info related to the new associations between new FG and SH blobs, and their correspondent KFs.
  - Delete and create new KFs if it is needed.
- KF Prediction of the new KF created: Time update KF.
- KF Correction: Measurement update.

7.4.1 Tracking using Kalman filters

The detected foreground objects and shadows are tracked using first order Kalman filters. The tracking and data association are based on a number of estimated parameters for the detected objects and shadows:

- The centroid of an ellipse fitting.
- The major and minor axis length of the ellipse.
- The probabilistic Appearance Model.

Each track is associated with these parameters, and a Kalman filter is used to predict the object's location using a first order motion model. Hence, the target state is defined by $x_t = (posx_t, posy_t, velx_t, vely_t, maj_t, min_t, \theta)$, which establishes a state vector for every observation, and adds the target speed and the size deformation rate at time t. Where $posx_t$ and $posy_t$ define the position (the centroid of the ellipse); $velx_t$ and $vely_t$ are velocity components; maj_t and min_t are the major and minor axis of the ellipse, respectively; and θ is the orientation. The KFs are initialised based on the detected foreground and shadow blobs, and the uncertainties are empirically estimated according to the precision of the detector.

7.4.2 Data association between blobs and Kalman filters

When performing data association five situations can occur:

i A new object: a new track is created.

ii A lost object: a track is destroyed if the object does not reappear within a certain number of frames (Tdead).

iii Object match: a one-to-one match, where the track is updated using the detected object assigned to it.

iv Object splitting: more than one detected object match a track. This is resolved by selecting the object with the highest probability of the matches, and creating new KFs for the other objects.

v Object merging: a single detected object matches two or more tracks, this is caused by inter-object occlusion, and is handled using probabilistic appearance models.

Data association algorithm

The foreground blobs extracted and classified as object or shadow (see section 7.3), are associated with a list of possible Kalman filters using Algorithm 7, which is based on the stable marriage algorithm [4].

Algorithm 7 Data Association between blob and KF

- **while** the list of blobs is not empty, **then**:
  - Evaluate the current blob (newblob).
  - **if** there is a KF associated to this blob, **then**:
    - **if** the best KF for this blob is not used, **then**:
      - **if** the position of the blob is close to the prediction and have similar appearance, **or** the blob position is far away but it has been lost previously and have similar appearance, **then**:
        - **Match** KF-newblob, KF Tstable.
      - **else** KF is invalid, **then**:
        - new KF.
        - **Match** newKF-newblob.
    - **else** KF is used, **then**:
      - **Get** the blob associated to the KF (oldblob).
      - **if** the newblob is more similar and has a better PAM match than the oldblob, **then**:
        - **Match** KF-newblob, KF Tstable.
        - **Free** KF-oldblob and add oldblob to blob-list.
      - **else**, **then**:
        - Check next best KF for this object.
  - **else** no KF is associated to this blob, **then**:
    - new KF.
    - **Match** newKF-newblob.
- **for** KFs not associated, **then**:
  - Lost KF, KF Tdead.

7.4.3 Probabilistic appearance models

Probabilistic appearance models inspired by [21] are applied for data association and to resolve inter-object occlusion. Each track has its own PAM, which consists of an RGB colour model with an associated probability mask. An example of a PAM is illustrated in figure 7.8. The colour model, which is denoted $M_{RGB}(\mathbf{x})$, shows the appearance of each pixel of an object. $P_c(\mathbf{x})$ denotes the probability mask and represents the probability of the object being observed at that pixel. The use of PAMs can be viewed as weighted template matching, where the template is $M_{RGB}(\mathbf{x})$ and the weights are given by $P_c(\mathbf{x})$. The coordinates of $\mathbf{x}$ are expressed using the coordinate system of the model, which is normalized to the object centroid. For each new track, a new PAM is created. In the object match situation, a track refinement step is applied before updating the model by finding the best fit in a small neighbourhood, e.g. 5×5 pixels. Track refinement increases

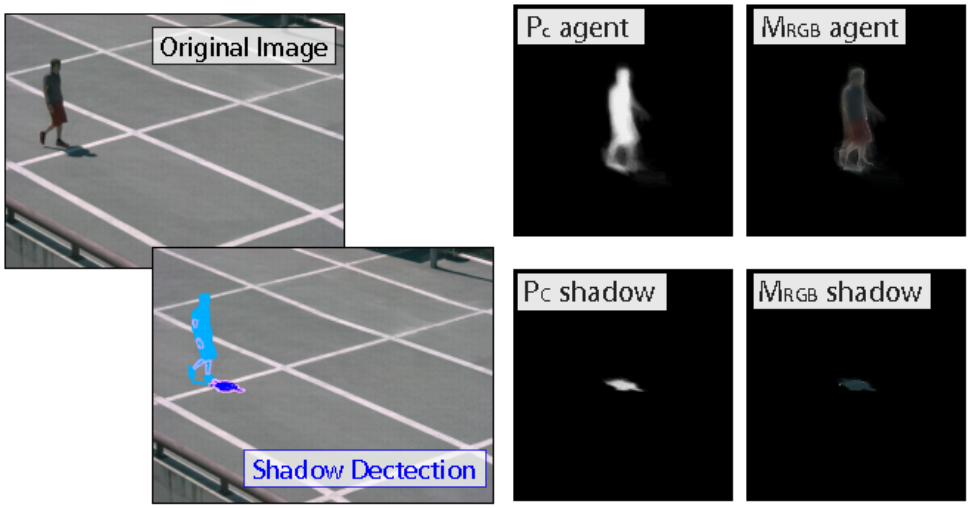


Figure 7.8: An example of a probabilistic appearance model, where the input Image, the shadow detection Image, the probability mask P_c and the color model M_{RGB} for the detected agent and shadow are shown.

the accuracy of the model; especially the colour model becomes sharper. When updating, the model usually stabilizes after less than 10 frames. Detail on building the model can be found in [21]. In the object merging situation the PAMs of the tracks are used to assign pixels of the detected object between the tracks using the estimated probability, as discussed in the following.

The foundation of the PAM is the ability to estimate the probability that a given pixel x of a detected object belongs to the model $\mathcal{M}_j$ of track j. This is denoted by $P\left(\mathcal{M}_j|I\left(\mathbf{x}\right)\right)$. I is the colour input image and is assumed to be normalized to the centroid of the detected object. The probability is calculated using Bayes' rule:

$$P\left(\mathcal{M}_j|I\left(\mathbf{x}\right)\right) \propto P_{RGB,j}\left(I\left(\mathbf{x}\right)|\mathcal{M}_j\right) \cdot P_{c,j}\left(\mathbf{x}\right) \tag{7.15}$$

The a priori probability is given by the probability mask of model $\mathcal{M}_j$, $P_{c,j}\left(\mathbf{x}\right)$, and $P_{RGB,j}\left(I\left(\mathbf{x}\right)|\mathcal{M}_j\right)$ is the color appearance likelihood, and this is approximated using a Gaussian color distribution:

$$P_{RGB,j}\left(I\left(\mathbf{x}\right)|\mathcal{M}_j\right) = \frac{1}{\left(2\pi\right)^{3/2}\left|\Sigma\right|^{1/2}} \cdot \tag{7.16}$$
$$\exp\left(-\frac{1}{2}\left(I\left(\mathbf{x}\right) - M_{RGB,j}\left(\mathbf{x}\right)\right)^T \Sigma^{-1}\left(I\left(\mathbf{x}\right) - M_{RGB,j}\left(\mathbf{x}\right)\right)\right)$$

The colour model for track j, $M_{RGB,j}$, represents the mean colour for each pixel. To reduce the complexity, the covariance matrix Σ can be assumed to be a diagonal matrix with identical variance σ in each colour channel. Given these assumptions Equation 7.16 reduces to:

$$P_{RGB,j}\left(I\left(\mathbf{x}\right)|\mathcal{M}_j\right) = \left(2\pi\sigma^2\right)^{-3/2} \cdot \exp\left(\frac{-\left\|I\left(\mathbf{x}\right) - M_{RGB}\left(\mathbf{x}\right)\right\|^2}{2\sigma^2}\right) \tag{7.17}$$

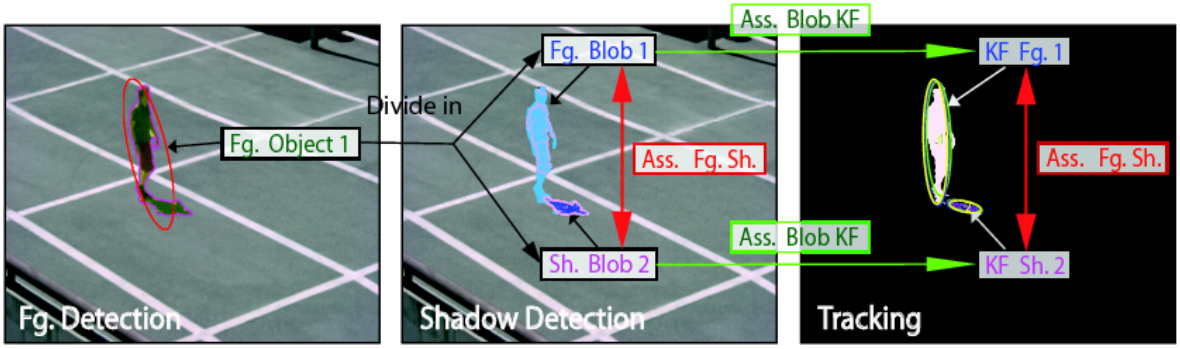


Figure 7.9: An example of the data association between FG, SH and the assigned KFs. The first image represents the FG detection. The second image shows the shadow detection, and how the analysed FG is divided into FG and SH blobs. In the third image the tracker has assigned one KF to each blob.

7.4.4 Object-shadow association

After the blobs (belonging to a FG or a SH) have been assigned to the KF, as described in sec. 7.4.2, the association between which shadow belongs to which FG and vice versa is saved in the KF info for use in the next frames. This information is used to identify the possible cases in the association between FG and SH. An example showing the data association between the blobs and the KFs, and how the data association between FG and SH is saved in the KF info, can be seen in Fig. 7.9. The first image of Fig. 7.9 represents the FG detection provided in section 7.3.1. The second image shows the shadow detection presented in sec. 7.4.2, and how the FG segmentation is further analysed and divided into FG and SH blobs. These blobs are associated, since both are part of the same FG object. In the third image the tracking system has assigned one KF to each blob, and the data association between the FG and the SH blobs is saved in the KF info.

7.4.5 Temporal consistency in the data association

The information related to the association between FG and SH saved in the KF is analysed, in order to check the possible data association cases, e.g., if a shadow has been lost. Fig. 7.7 shows an example of how the approach works in the case, when a miss-detected shadow is recovered by the shadow tracking. The shadow tracking (and the figure) is explained further in the following.

When testing for temporal consistency in the data association between FG and SH with their respective KF, three situations can occur:

- FG and SH match:
  The association created at time t-1 continues at time t, which is the ideal case.
- New shadow: FG/SH splitting.
  A new association between the FG and SH is created at time t. Temporal association of a splitting shadows is applied, in order to avoid miss-detected shadows in posterior frames.

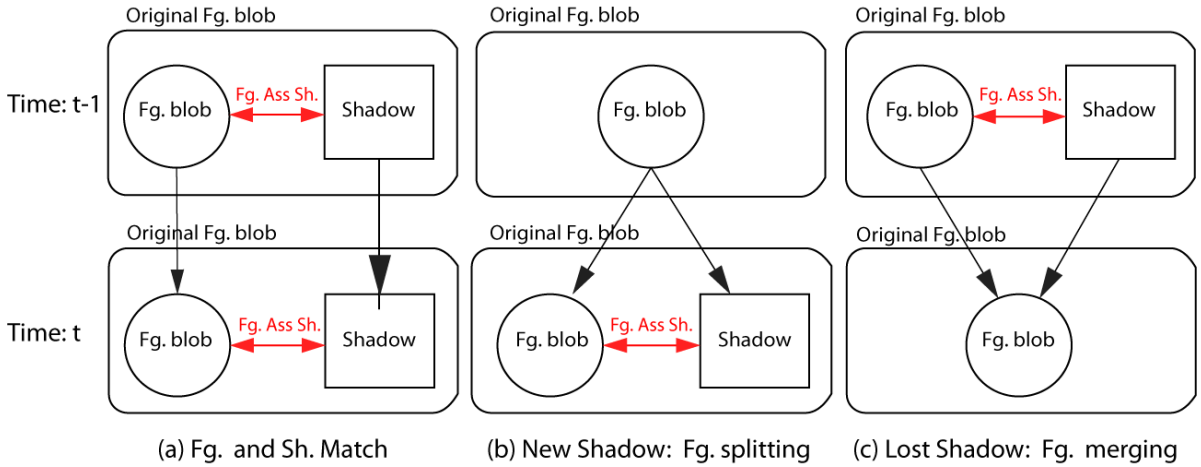


Figure 7.10: The three possible data association situations between FG, SH and their KFs. A rounded rectangle illustrates an original FG blob before shadow detection, a circle illustrates a FG and a square illustrates a SH from the shadow detection. A double red arrow indicates an association between FG and SH, and a black arrow indicates an association between FG and SH in the next frame.

- Lost shadow: FG-SH merging.
  The association between the FG and SH at time t-1 is lost at time t, since the shadow is miss-detected (Fig. 7.7).

The three cases are illustrated in the Fig. 7.10. It is possible that a new shadows appears or a shadow is lost, without a splitting or merging of the FG object. However, these cases are not of interest, since they do not have any FG-SH association, hence, they will be tracked in an usual manner by the KFs. A shadow is considered lost when the blob (the KF that is associated with this blob) fulfill a set of conditions: it was classified as SH at time t-1 (the previous frame), and it had a FG associated, which also had this SH associated. At time t (the current frame), this FG has no shadow associated and the SH has also lost the association with this FG, then the shadow is considered lost. The shadow region can be recovered by evaluating the FG blob (which contains the merged FG and SH), the blob prediction for the FG KF, and the blob prediction for the lost SH KF. The next subsections explains the recovery process for the lost shadow case.

Recovering lost shadows

The FG blob which belongs to the FG KF, associated in the previous frame with the shadow considered lost, is analysed in order to recover the possible shadow region. To do so, the mask of the positives edges (*Edpos* mask) plus morphological operators are applied, to divide the FG blob into FGs and possible shadow regions. Multiple regions can be found but theoretically only one is the shadow. This happens because the positive edges are used to divide the image, and these edges come from the current image. As explained in sec. 7.2, one of the characteristics of shadows is that they can only have negative edges, i.e., the edges from the background image. Therefore, theoretically several FG

regions can be found but only one SH region. In Fig. 7.7 it can be seen how the original FG blob detected, as described in section 7.3, is subdivided into the possible chromatic shadow regions (image Chr.Sh.Regions in the figure; the regions in the image are shown in different colours) using the *Edpos* mask. In the following, the new divided blobs of the regions are associated with the predictions of the KFs, in order to recover the chromatic shadows.

Correspondence matrix for the new divided blobs

The weights of the blob prediction for the FG KF and the SH KF are calculated w.r.t. all possible regions found in the previous step. Therefore, two correspondence matrix are computed, where one contains the euclidean distance between the new blobs and the FG and SH KF predictions, and the other the overlapping (matching) between the new blobs and the FG and SH KF predictions. Next, these weights are applied to associate the SH and FG KF predictions with the blobs.

Association between KF predictions and the new blobs

The best match (shortest distance and best overlap) between the SH KF predictions and the blob will be considered as the shadow region, while the other blobs will be considered as FG blobs, since only one region can be shadow. Hence, the other blobs have to be FGs. In this way, by using the tracking information, the original FG blob can be segmented into FG and SH regions, and thereby recover miss-detected chromatic shadows. This information is used as a feedback from the tracking to the shadow detection step. Fig. 7.7 shows how the blobs extracted from the divided regions are associated with the prediction of the KFs, in order to detect the chromatic shadow.

7.4.6 Feedback to the chromatic shadow detection

Once the chromatic shadows are detected, the original image (original FG blob) is divided into only one blob for the FGs and only one blob for the SHs, using the information from the positive edges and the shadow tracking. Hence, the FG blob will be associated with the FG KF and the SH blob will be associated with the lost SH KF. Next, the original image is updated, so the miss-detected chromatic shadows are now marked as detected. Fig. 7.7 shows how the detected chromatic shadow is updated according to the original blob after the feedback from the shadow tracking.

7.4.7 Managing and updating KFs and PAMs

Finally, the information related to the new associations between the FG and SH blobs, and their respective KFs have to be updated. Additionally, the KFs have to be updated with the new associated blobs, and the PAMs have to be updated considering the new blobs. It is possible that a new KF is erroneously created because one object together with its shadow were considered as a new object. Therefore, the new KFs created in the

Figure 7.11: An original image from the Outdoor_Cam1 sequence, and foreground results after shadow removal using the Huerta et al. approach [7], the Zivkovic et al. approach [24] using a shadow detector [17] and our approach, respectively.

data association between the blobs and the KFs have to be checked. Any unused KFs, which were assigned to blobs considered as lost shadows, are deleted.

Consequently, due to the data association between FG and SH we have achieved: (i) enhancing the chromatic shadow detection by recovering miss-detecting shadows, which were incorrectly detected by the shadow detector (sec. 7.3). (ii) Improving the segmentation for high level processes, such as detection and tracking, by avoiding shadows. (iii) A more robust tracking by exploiting FG and SH association. In Fig. 7.7 an example is given, where the output of the tracking process is shown with and without our Top-down approach (the last two images at the right called "Tracker End"). The figure of the Top-down approach shows how the system recovers detecting the chromatic shadow, hence, the FG KF and SH KF are correctly updated. On the other hand, the image of the tracker, without taking the association of the FG, SH and their assigned KFs into account, shows how the FG and SH KF are lost and a new false KF is created.

7.5 Experimental Results

The results presented in this section are from tests conducted on datasets selected from well-known databases[2] and our own recordings. Our approach is tested on sequences of outdoor and indoor scenarios, and compared to other statistical approaches when results are available. The chosen test sequences are relatively long and umbra and penumbra shadows are cast by multiple foreground objects. The sequences analysed are Outdoor_Cam1 (800 frames, 607x387 pixels), HigwayIII[1] (2227 frames, 320x240 pixels), HallwayI[1] (1800 frames, 320x240 pixels) and HERMES_ETSEdoor_day21_I4 (6500 frames, 640x480 pixels).

Figure 7.11 and 7.12 show the results when comparing our shadow detector with state-of-the-art approaches [10, 17, 12, 7, 24]. As it can be seen in these figures our approach outperforms the other analysed methods. However, in a few cases the gradient masks cannot be accurately build due to camouflage and noise problems. Thus, the separation of a foreground object and a shadow region can fail. Occasionally, when the anomaly in case 2-2 (see sec. 7.2.3) occurs and a part of the foreground object or the shadow is not segmented due to segmentation problems, the shadow detection can miss-classify the shadow as a foreground object (see Fig. 7.14.c and 7.14.h. The top-down shadow tracking approach can solve some of these problems, as it can be seen in Fig. 7.14.d and 7.14.i.

[2] http://vision.gel.ulaval.ca/~CastShadows/

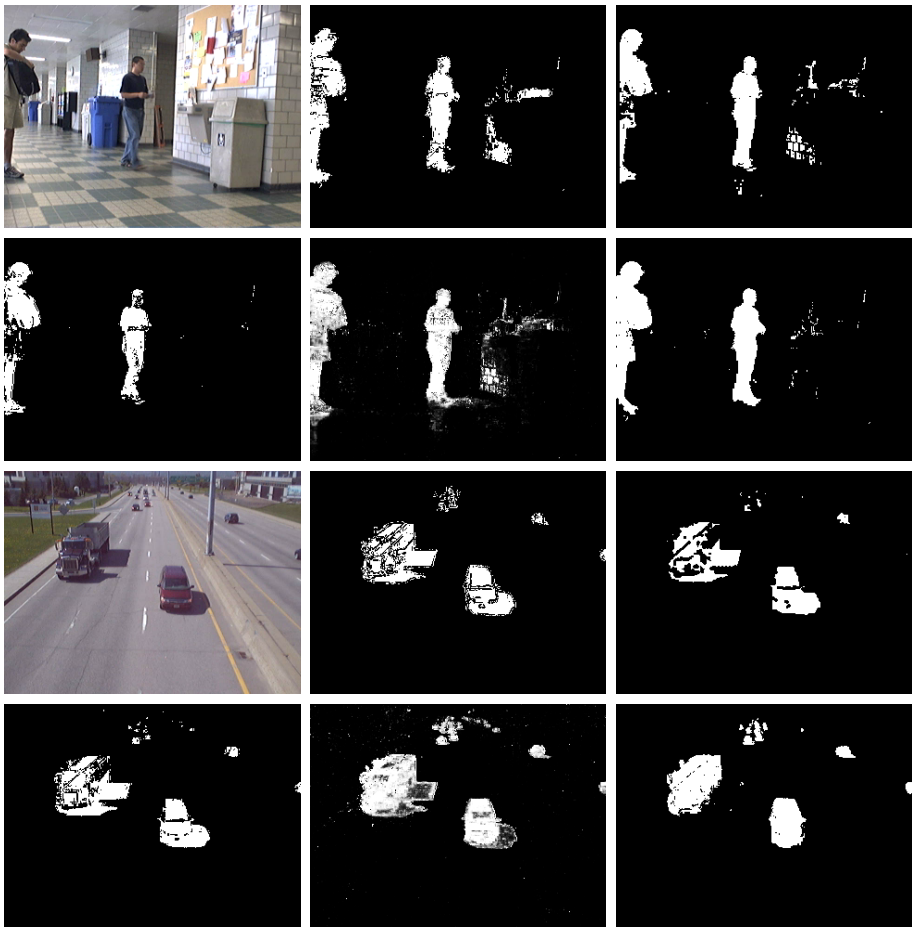


Figure 7.12: Original images from the HallwayI and HighwayIII sequences, and foreground results after shadow removal using the Huerta et al. approach [7], the Kim et al. approach [10], the Zivkovic et al. approach [24] using a shadow detector [17], the Martel et al. approach [12] and our approach, respectively (read row-wise).

To evaluate our approach in a quantitative way, it is compared with the approaches in [12, 11] using the most employed quantitative expressions utilized to evaluate the shadow detection performance: the Shadow Detection Rate (SR) and the Shadow Discriminate Rate (SD). Refer to [17] for the exact equations. The results in Table 7.1 shows that our method outperforms both the parametric approach based on Gaussian mixtures GMSM [11] and the nonparametric physical model [12]. Note that the results for the GMSM [11] and the physical model [12] on the Hallway sequence have been obtained directly from [12]. Additionally, it should be noted that our approach needs a reasonable resolution to work correctly. Furthermore, shadow regions need to have a minimum area for analysis, or there might not be enough information for a proper shadow detection and classification. Fig. 7.13 shows examples of the foreground and shadow detection.

The top-down process assists the chromatic shadow detector when it fails to detect shadows, as shown in Figure 7.14.c and 7.14.h. The tracking system is able to track the shadows, and use this information as feedback to the chromatic shadow detector. Hence, the miss-detected shadows can be recovered and correctly detected. Figure 7.14

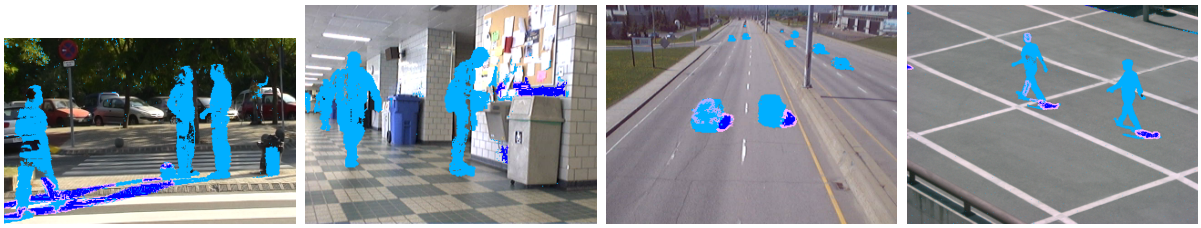


Figure 7.13: Chromatic Shadow detection results for the CVC_Outdoor_Cam1, LVSN_HallwayI, LVSN_HighwayIII and HERMES_ETSEdoor_day21 sequences using our shadow detector. Foreground segmentation results are coloured in cyan, and the shadow detection results are coloured in blue.

Method	HallwayI		HERMES ETSEdoor_day21_I4	
	SR	SD	SR	SD
GMSM	0.605	0.870	–	–
Physical model	0.724	0.867	–	–
Bottom-up	0.807	0.907	0.363	0.983
Top-down	–	–	0.417	0.978

Table 7.1: SR and SD results for our approach (bottom-up: chromatic shadow detection and top-down: shadow tracking) and two other successful methods: Gaussian Mixture Shadow Models (GMSM) [11] and a physical model of light and surfaces [12].

and 7.15 present examples of shadow recovery using our top-down approach for the LVSN_HighwayIII and the HERMES_ETSEdoor_day21 sequences. Fig. 7.14.a and 7.14.f show the foreground detection results achieved by our combined approach for the LVSN_HighwayIII and HERMES_ETSEdoor_day21_I4 sequence, respectively. In Fig. 7.14.b and 7.14.g the chromatic shadow detection results of our detector are shown. Note that the shadows are not correctly detected. Fig. 7.14.c and 7.14.h show the output of the tracker without applying our top-down approach, while Fig. 7.14.d and 7.14.i show the results of our top-down approach. Finally, Fig. 7.14.e and 7.14.j show how the chromatic shadows are accurately detected after the feedback from the tracker to the chromatic shadow detector.

For the LVSN_HighwayIII sequence shown in Figure 7.14 our top-down approach is able to detect the chromatic shadows However, this scenario is very difficult to track, since the objects move very fast compared to the frame rate of the sequence. Additionally, the appearance of the objects changes very quickly. In this case the tracks are sometimes lost, and therefore it is not possible to run the top-down process throughout all of the sequence.

The input image from the HERMES_ETSEdoor_day21_I4 sequence in Figure 7.14 is captured without a detected shadow for 10 frames. Hence, this example illustrates (Fig. 7.14.h) how the tracker's KF assigned to the shadow is completely lost without the top-down approach, while the other tracker's KF is tracking the combined FG and SH blob (a Red ellipse depicts the a posteriori state of the KF). In some cases the tracker will

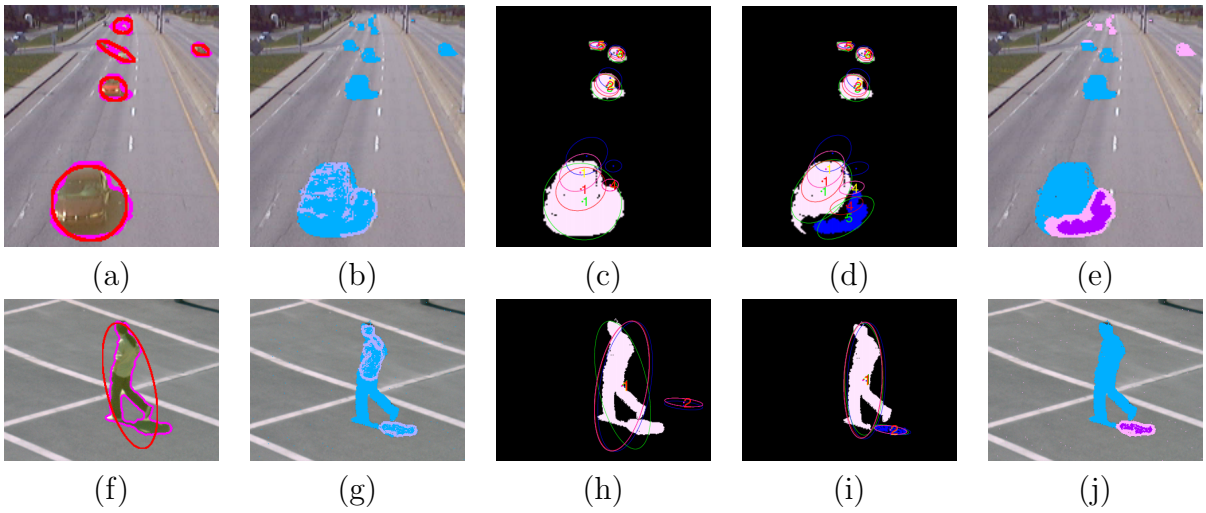


Figure 7.14: Examples of shadow recovery using our top-down approach for the LVSN_HighwayIII and HERMES_ETSEdoor_day21_I4 sequences. (a) and (f) foreground detection images. (b) and (g) chromatic shadow detection results of our detector. Note that the shadows are not correctly detected. (c) and (h) output of the tracker without applying our top-down approach. The KFs associated to the shadows are lost and therefore falsely updated. (d) and (i) output of the tracker, after the chromatic shadows are recovered, using our top-down process. The shadows are accurately detected, therefore the FG KFs and the SH KFs are correctly updated, and none of them are lost. The a posteriori state of the tracker is depicted with a red ellipse. Image (e) and (j) shows the final chromatic shadow detection results in the original image (the shown image examples are cropped).

create a new KF, since the combined FG and SH blob is so different that the system thinks it is a new object. In contrast, Fig. 7.14.i shows the output of the tracker using our top-down process. In this image the shadow is accurately detected, and the KFs are correctly updated. This is illustrated by the red ellipses in the image. Fig. 7.15 shows a number of processed frames, depicting the results using our top-down approach. In the figure it can be seen how our approach is able to track the objects and the shadows, and when the chromatic shadow is lost, the system is able to recover it.

The quantitative results in Table 7.1 for the HERMES_ETSEdoor_day21_I4 sequence show how the shadow detection rate (SR) is improved from 0.363 to 0.417, which effectively means that we detect more shadow regions using shadow tracking, while The shadow discriminate rate (SD) is pretty stable: 0.983 and 0.978. Hence, the foreground is still robustly segmented when shadows are tracked.

7.6 Conclusion

In this paper, we have presented two main novelties: (i) a bottom-up approach for detection and removal of chromatic moving shadows in surveillance scenarios, and (ii) a top-down approach based on Kalman filters to detect and track object and shadows to

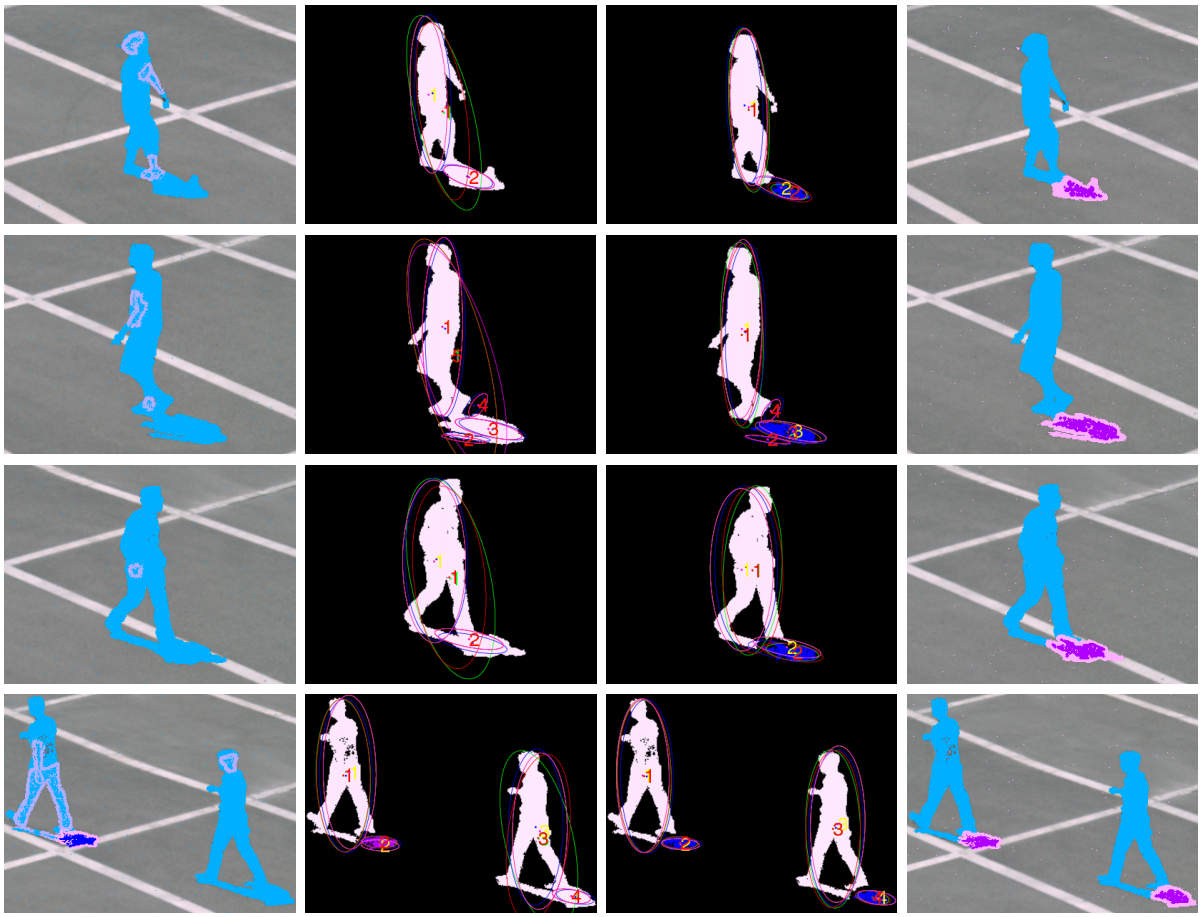


Figure 7.15: Chromatic Shadows detection results using our top-down approach for the HERMES_ETSEdoor_day21_I4 sequence. First column shows the chromatic shadow detection results. Note that the shadows are not correctly detected. Second column shows the output of the tracker without the association between FG-SH, where the tracks for the shadows are lost. Third column shows the tracker output using our top-down approach. The chromatic shadows are detected and the tracker are correctly updated for the FG and the SH. The a posteriori state of the tracker is depicted with a red ellipse. The last column shows how the chromatic shadow is recovered and correctly detected in the original image (the shown image examples are cropped).

enhance the chromatic shadow detection. The Bottom-up part the shadow detection approach apply a novel technique based on gradient and colour models for separating chromatic moving shadows from moving objects. Firstly, we extend and improve well-known colour and gradient models into an invariant colour cone model and an invariant gradient model, respectively, to perform automatic segmentation, while detecting potential shadows. Hereafter, the regions corresponding to potential shadows are grouped by considering "a bluish effect" and an edge partitioning. Lastly, (i) temporal similarities between local gradient structures and (ii) spatial similarities between chrominance angle and brightness distortions are analysed for all potential shadow regions, in order to finally identify umbra shadows.

The resulting shadow detection can detect and remove chromatic moving shadows (umbra shadows) and penumbra shadows, while several other methods are restricted to the latter. However, in some cases the separation between a foreground object and a shadow region can fail. Occasionally, a part of the foreground object or the shadow is not accurately segmented due to segmentation problems, e.g., camouflage. Therefore, the shadow detection can miss-classify a shadow as being a part of a foreground object. In order to solve this problem a top-down approach has been developed, which tracks both objects and shadows using Kalman filters. Consequently, due to the data association between FG and SH we have achieved: (i) enhancement of the chromatic shadow detection by recovering miss-detected shadows. (ii) A more robust tracking by using mutual information and association of object and shadow. (iii) Improvement of the segmentation for high level processes, such as detection and tracking, by avoiding shadows.

Qualitative and quantitative results of tests for both outdoor and indoor sequences from well-known databases validate the presented approach. Overall, our approach gives a more robust and accurate shadow detection and foreground segmentation compared to the state-of-the-art methods. Unlike other approaches, our method does not make any a-priori assumptions about camera location, surface geometries, surface textures, shapes and types of shadows, objects, and background.

However, some remarks have to be made with respect to the bottom-up part (chromatic shadow detector) and the top-down part (shadow tracking). The chromatic shadow detector needs a reasonable resolution to work correctly, and noisy and blurred images intensify the camouflage problems. Furthermore, shadow regions need to have a minimum area for analysis, or there might not be enough information for a proper shadow detection and classification. The "bluish effect" gives very good results for some outdoor sequences. However, sometimes it does not work as defined theoretically, since it is affected by external factors, such as the sensibility of the camera and image compression. For the tracking process, targets are assumed to move with a reasonable velocity compared to the frame rate, since objects which move quickly with rapidly appearance changes can cause tracking difficulties.

Acknowledgment

This work is supported by the EC grants IST-027110 for the HERMES project; IST-045547 for the VIDI-Video project; by the Spanish MEC under projects TIN2006-14606 and CONSOLIDER-INGENIO 2010 MIPRCV CSD2007-00018; and the Danish National Research Councils - FTP under the research project "Big Brother *is* watching you!".

References

[1] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *TPAMI*, 25(10):1337–1342, 2003.

[2] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti. Improving shadow suppression in moving object detection with hsv color information. In *Intelligent Transportation Systems*, 2001.

[3] G.D. Finlayson, S.D. Hordley, C. Lu, and M.S. Drew. On the removal of shadows from images. *TPAMI*, 28(1):59–68, 2006.

[4] D. Gusfield. The stable marriage problem: structure and algorithms. *MIT Press*, 1989.

[5] M. Heikkila and M. Pietikainen. A texture-based method for modeling the background and detecting moving objects. *TPAMI*, 28(4):657–662, 2006.

[6] T. Horprasert, D.Harwood, and L.S.Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Frame-Rate Applications Workshop*, 1999.

[7] I. Huerta, A. Amato, J. Gonzàlez, and J.J. Villanueva. Fusing edge cues to handle colour problems in image segmentation. *AMDO*, 5098:279–288, 2008.

[8] I. Huerta, M. Holte, T.B. Moeslund, and J. Gonzàlez. Detection and removal of chromatic moving shadows in surveillance scenarios. In *ICCV*, 2009.

[9] Y. Ivanov, A. Bobick, and J. Liu. Fast lighting independent background subtraction. *IJCV*, 37(2):199–207, 2000.

[10] K. Kim, T.H. Chalidabhongse, D. Harwood, and L.S. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3):172–185, 2005.

[11] N. Martel-Brisson and A. Zaccarin. Learning and removing cast shadows through a multidistribution approach. *TPAMI*, 29(7):1133–1146, 2007.

[12] N. Martel-Brisson and A. Zaccarin. Kernel-based learning of cast shadows from a physical model of light sources and surfaces for low-level segmentation. In *CVPR*, 2008.

[13] T. B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 104:90–126, 2006.

[14] S. Nadimi and B. Bhanu. Physical models for moving shadow and object detection in video. *TPAMI*, 26(8):1079–1087, 2004.

[15] K. Onoguchi. Shadow elimination method for moving object detection. In *ICPR*, 1998.

[16] F. Porikli and J. Thornton. Shadow flow: a recursive method to learn moving cast shadows. In *ICCV*, 2005.

[17] A. Prati, I. Mikic, M. Trivedi, and R. Cucchiara. Detecting moving shadows: Algorithms and evaluation. *TPAMI*, 25(7):918–923, 2003.

[18] R.O'Callaghan and T. Haga. Robust change-detection by normalised gradient-correlation. In *CVPR*, 2007.

[19] E. Salvador, A. Cavallaro, and T. Ebrahimi. Cast shadow segmentation using invariant color features. *CVIU*, 95(2):238–259, 2004.

[20] O. Schreer, I. Feldmann, U. Goelz, and P. Kauff. Fast and robust shadow detection in videoconference applications. In *VIPromCom*, 2002.

[21] A. Senior, A. Hampapur, Y.-L. Tian, L. Brown, S. Pankanti, and R. Bolle. Appearance models for occlusion handlinge. *IVC*, 24(11):1233–1243, 2006.

[22] J. Stauder, R. Mech, and J. Ostermann. Detection of moving cast shadows for object segmentation. *Trans. Multimedia*, 1(1):65–76, 1999.

[23] Yair Weiss. Deriving intrinsic images from image sequences. In *ICCV*, 2001.

[24] Z. Zivkovic and F. Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780, 2006.

Chapter 8

Conclusion

In this chapter a discussion and concluding remarks on the findings of this Ph.D. study will be given. This Ph.D. study has lead to several publications, where six of them are included directly in this thesis. The presented work targets and deals with several important aspects of human activity recognition: different kinds of activities, 2D and 3D techniques, simple constrained and complex unconstrained scenes, global and local feature representations, a review of recent developments in the field, foreground segmentation and shadow detection.

Robust methods for recognizing different kind of activities have been developed, from one and two arms gestures (*e.g.* point, wave, clap etc.) to full-body actions (*e.g.* walk, run, jump etc.). The activities have been recorded by standard color cameras, multi-view camera setups and time-of-flight (ToF) range cameras, enabling analysis of both 2D and 3D video data. The 2D image data recorded by standard color cameras captures both activities performed in simple scenes with controlled settings (*i.e.* one actor, steady camera, simple and clean background, and low variation in scale, rotation, viewpoint and illumination) and complex scenes with unconstrained settings (*i.e.* multiple actors, moving camera, background clutter, and high variation in scale, rotation, viewpoint and illumination).

Activity recognition in simple scenes with controlled settings, like laboratory environments or carefully designed scenes can provide valuable information about the performance of methods and systems. However, the variability and challenges of real-life scenes are not investigated in this way, often leading to less general and less applicable methods. Hence, the the presented work also focuses on scenes that are not carefully constrained. Furthermore, The use of both global and local image features have been investigated for activity recognition

The presented work on 2D action recognition specifically targets dynamic outdoor scenes and addresses multiple people. For this purpose an approach based on detection of spatio-temporal interest points (STIPs) and local description of image features has been developed, where robust and selective STIPs are detected by applying surround suppression combined with local and temporal constraints. The approach is especially robust to camera motion and background clutter, where other detectors fails, and improves the performance by detecting more repeatable, stable and distinctive STIPs for human ac-

tors, while suppressing unwanted background STIPs. Using this approach we have shown state-of-the-art performance on several challenging and benchmark datasets.

Although the research within vision-based human activity recognition has come far the past decade, which has lead to development of systems capable of simple action recognition in constrained scenes and controlled settings, the problem of robust recognition of daily activities in complex scenes with several challenges, like scale-, rotation and viewpoint changes, difficult lighting conditions, moving camera, background clutter, occlusion and multiple people, is far from being solved. This can especially be seen on the results reported for the challenging Hollywood 2 dataset of human actions from selected movies, where state-of-the-art recognition accuracy is still pretty low (58.45%).

For acquisition of 3D data both direct 3D imaging devices (ToF range cameras) and 3D reconstruction from multiple camera views are applied, to explore the challenges of different quality of 3D data and the advantages of each technology. To this end, 3D approaches for robust human activity recognition using both type of sensors have been presented. The approaches are based on extended concepts of the developed 2D techniques for feature extraction and classification strategies. Especially, the local 3D motion feature description from 4D multi-view STIPs has shown state-of-the-art performance on publicly available datasets.

Regarding the work on multi-view camera systems, a review and comparative study of recent developments for human 3D body modeling, pose estimation and activity recognition using multi-view data has been provided to give the reader an overview of proposed approaches, seen with respect to the different application areas and their associated requirements for successful operation.

The main strength of multi-view setups is the high quality full-volume 3D data, which can be provided from 3D reconstruction by shape-from-silhouettes and refinements techniques. It also helps to uncover occluded action regions from different views in the global 3D data, and allows for extraction of informative features in a more rich 3D space, than the one captured from a single view. However, although the reviewed approaches show promising results for multi-view human body modeling, pose estimation and action recognition, 3D reconstructed data from multi-view camera systems has some shortcomings. First of all, the quality of the silhouettes is crucial for the outcome of applying shape-from-silhouettes. Hence, shadows, holes and other errors due to inaccurate foreground segmentation will affect the final quality of the reconstructed 3D data. Secondly, the number of views and the image resolution will influent the level of details which can be achieved, and self-occlusion is a known problem when reconstructing 3D data from multi-view image data, resulting in merging body parts. Finally, 3D data can only be reconstructed in a limited space where multiple camera views overlap.

In recent years other prominent vision-based sensors for acquiring 3D data have been developed, like ToF range cameras. Especially, with the introduction of the Microsoft Kinect sensor, these single and direct 3D imaging devices have become widespread and commercial available at low cost. Although these sensors only captures 3D data of the frontal surfaces of humans and other objects, their applicability are much broader due to the convenience of using a single sensor, avoiding the difficulties inherent to classical stereo and multi-view approaches (the correspondence problem, careful camera placement

and calibration). Additionally, the need for multiple calibrated and synchronized cameras is obviously not desirable. Hence, the future of acquiring vision-based 3D data will move in this direction, and in the next years we will see many new proposed approaches for human body modeling, pose estimation, activity recognition and other computer vision related topics using these sensors. In future work it would be interesting to adapt the multi-view approach based on local 3D motion feature description from 4D STIPs to single view depth sensors, like Kinect.

Finally, an approach for automatic foreground segmentation and shadow detection has been designed. Foreground segmentation is one of the most used preprocessing steps for many computer vision algorithms to extract regions of interest, *e.g.* for activity recognition, while the impact of shadows is a notorious problem in computer vision. The work explores a multi-stage approach for foreground segmentation and shadow detection. Firstly, a bottom-up architecture using a novel technique based on gradient and color models has been presented for separating chromatic moving cast shadows from detected moving objects. Secondly, a top-down architecture based on a tracking system using mutual object-shadow information has been developed, in order to enhance the chromatic shadow detection.

Although, detection and removal of shadows is important for successful and precise foreground segmentation, and the approach has shown promising performance on publicly available datasets, the problem of shadow detection is still far from being solved. The proposed chromatic shadow detector needs a reasonable resolution to work correctly, and noisy and blurred images intensify the camouflage problems. Furthermore, shadow regions need to have a minimum area for analysis, or there might not be enough information for a proper shadow detection and classification. The "bluish effect" gives very good results for some outdoor sequences. However, sometimes it does not work as defined theoretically, since it is affected by external factors, such as the sensibility of the camera and image compression. For the tracking process, targets are assumed to move with a reasonable velocity compared to the frame rate, since objects which move quickly with rapidly appearance changes can cause tracking difficulties. The proposed tracking system performs well for basic scene situations. However, for more complex scenario, such as crowded scenes or situations with multiple grouping and splitting processes, the tracks are sometimes lost and the tracker information becomes corrupted. Thus, in future work a high level tracker is needed, which can manage the low level trackers in a top-down architecture.